

ADA 102738

RADC-TR-81-103
Phase Report
June 1981

LEVEL *X*

(12)



SURVEY OF NUMERICAL METHODS FOR SOLUTION OF LARGE SYSTEMS OF LINEAR EQUATIONS FOR ELECTROMAGNETIC FIELD PROBLEMS

Rochester Institute of Technology

Tapan K. Sarkar
Kenneth R. Siarkiewicz (RADC)
Roy F. Stratton (RADC)



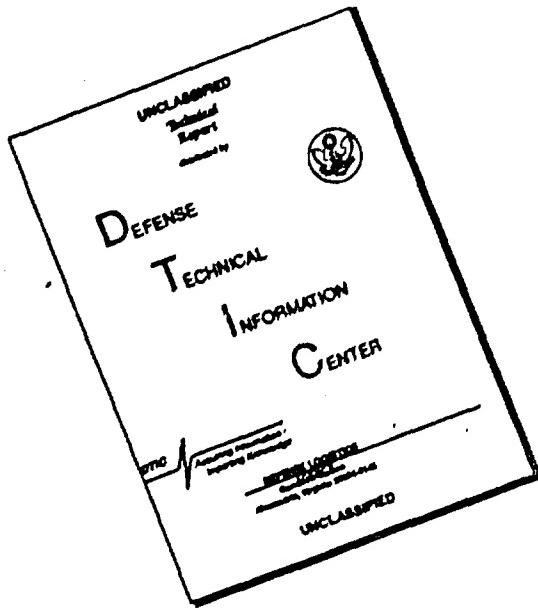
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DIGITAL FILE COPY

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

818 13014

DISCLAIMER NOTICE



**THIS DOCUMENT IS BEST
QUALITY AVAILABLE. THE COPY
FURNISHED TO DTIC CONTAINED
A SIGNIFICANT NUMBER OF
PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-81-103 has been reviewed and is approved for publication.

APPROVED:



KENNETH R. SIARKIEWICZ
Project Engineer

APPROVED:



DAVID C. LUKE, Colonel, USAF
Chief, Reliability & Compatibility Division

FOR THE COMMANDER:



JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (RBCT) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

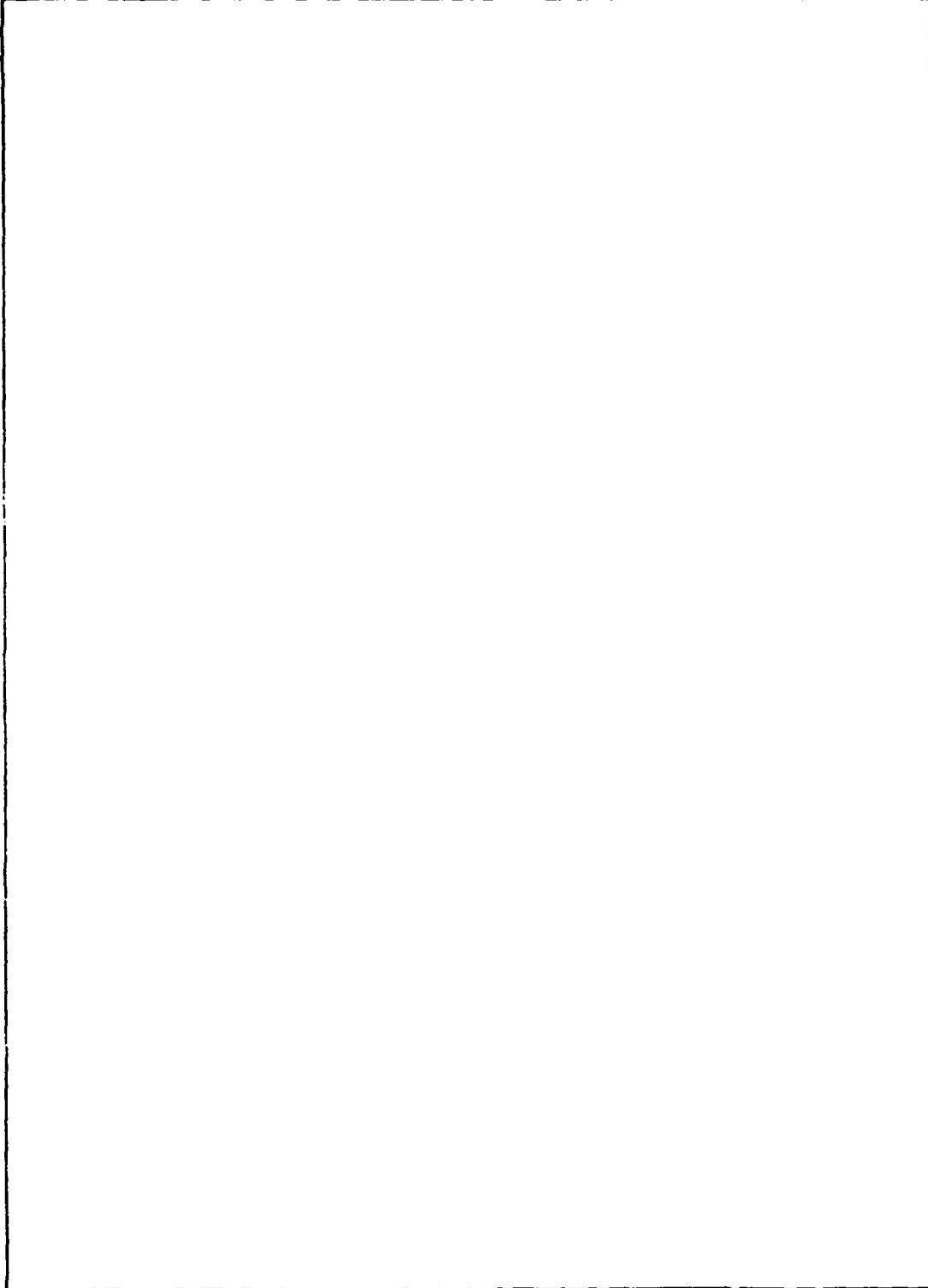
UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC/TR-81-103	2. GOVT ACCESSION NO. AD-A102738	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SURVEY OF NUMERICAL METHODS FOR SOLUTION OF LARGE SYSTEMS OF LINEAR EQUATIONS FOR ELECTROMAGNETIC FIELD PROBLEMS.		5. TYPE OF REPORT & PERIOD COVERED Phase Report Nov 78 - Sep 79
7. AUTHOR(s) Tapan K. Sarkar Kenneth R. Siarkiewicz * Roy F. Stratton *		6. PERFORMING ORG. REPORT NUMBER N/A
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Electrical Engineering Rochester Institute of Technology Rochester NY 14623		8. CONTRACT OR GRANT NUMBER(S) F30602-78-C-0083
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RBCT) Griffiss AFB NY 13441		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 233803PG
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		12. REPORT DATE June 1981
		13. NUMBER OF PAGES 114
		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES * Rome Air Development Center RADC Project Engineer: Kenneth R. Siarkiewicz (RBCT)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Electromagnetic Fields Numerical Analysis Linear Equation Solution Techniques Method of Moments Matrix Equation Solution Techniques		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The objective of this report is to survey many of the popular methods for the solution of large matrix equations with the hope of finding an efficient method suitable for both electromagnetic scattering and radiation problems and system identification problems.		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

NOTATIONS

A Represents a MxM square matrix

X Y Are Mxl matrices

A^{ij} Represents the element belonging to the ith row and jth column

xⁱ Is the ith element of X

X_n Represents the matrix obtained after n iterations in iterative methods and in direct methods it is just another matrix obtained after processing it n times.

$\|X\|$ Represents the norm of X

cond [A] Is the condition number of A $\triangleq \frac{\text{largest eigenvalue of } (A)}{\text{minimum eigenvalue of } (A)}$

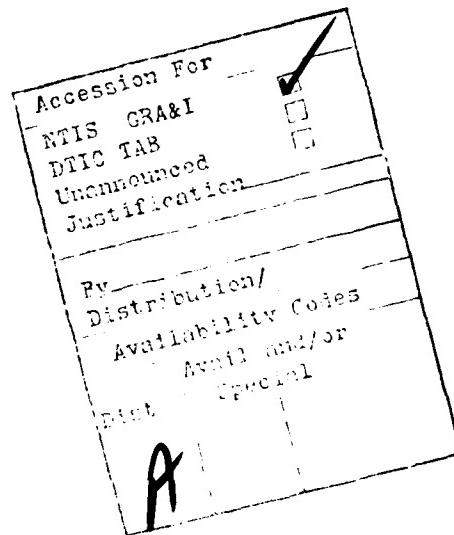


TABLE OF CONTENTS

1. INTRODUCTION	1
2. DIRECT METHODS FOR SOLVING MATRIX EQUATIONS	3
2.1 CRAMER'S RULE (THE ADJOINT METHOD)	4
2.2 GAUSSIAN ELIMINATION AND LU DECOMPOSITION	5
2.3 COMPACT METHODS (CROUT AND DOLITTLE OR CHOLESKY'S METHOD)	8
2.4 IS GAUSSIAN ELIMINATION REALLY OPTIMUM?	10
2.5 ANALYSIS OF ROUND-OFF ERRORS FOR DIRECT METHODS OF SOLVING A SYSTEM OF LINEAR EQUATIONS	11
2.6 CONCLUSIONS	21
3. PHILOSOPHY OF ITERATIVE METHODS FOR SOLVING MATRIX EQUATIONS	22
3.1 MATHEMATICAL APPROACH OF ITERATIVE METHODS	23
4. LINEAR ITERATIVE METHODS	28
4.1 GAUSS'S HAND RELAXATION METHOD	29
4.2 JACOBI'S CYCLICAL ITERATION METHOD (SIMULTANEOUS DISPLACEMENT METHOD)	29
4.3 SEIDEL'S METHOD (SUCCESSIVE DISPLACEMENT METHOD)	30
4.4 BACK AND FORTH SEIDEL PROCESS	33
4.5 THE METHOD OF SUCCESSIVE OVER/UNDER RELAXATION	35
5. MONTE CARLO METHOD	38
5.1 SOLUTION OF A SYSTEM OF LINEAR EQUATIONS	39
5.2 ERRORS IN MONTE CARLO METHOD	47
6. COMPARISONS OF EFFICIENCIES BETWEEN MONTE CARLO METHOD, GAUSSIAN ELIMINATION AND LINEAR ITERATIVE SCHEMES	48
6.1 DERIVATION OF COMPUTATIONAL REQUIREMENTS	49
7. NONLINEAR ITERATIVE SCHEMES	55
7.1 HISTORY OF NONLINEAR ITERATIVE SCHEMES	56
7.2 METHOD OF STEEPEST DESCENT	57

7.3 CONJUGATE DIRECTION METHOD	59
7.4 CONJUGATE GRADIENT METHOD	60
8. ANALYSIS OF CONVERGENCE OF VARIOUS ITERATIVE SCHEMES	64
8.1 RATE OF CONVERGENCE FOR LINEAR ITERATIVE SCHEMES	65
8.2 RATE OF CONVERGENCE FOR NONLINEAR ITERATIVE SCHEMES.	69
8.2.1 METHOD OF STEEPEST DESCENT	69
8.2.2 METHOD OF CONJUGATE GRADIENT	72
8.2.3 THE J STEPS STEEPEST DESCENT METHOD.	76
9. ROUND-OFF ERRORS ASSOCIATED WITH ITERATIVE SCHEMES.	78
9.1 ERROR ANALYSIS	79
10. EXTENSION OF DIRECT AND ITERATIVE METHODS TO COMPLEX UNSYMMETRIC MATRICES.	82
11. MINIMIZATION OF THE CONDITION NUMBER OF A MATRIX FOR ACCELERATING ITERATIVE METHODS AND REDUCING ROUND-OFF ERRORS IN DIRECT METHODS .	84
11.1 DERIVATION OF THE OPTIMUM ACCELERATION PARAMETER.	85
12. CORE STORAGE REQUIRED FOR VARIOUS METHODS	89
13. WORK REQUIRED FOR VARIOUS METHODS	90
14. A SPECIAL NOTE ON THE CONJUGATE GRADIENT METHOD	92
15. SUMMARY AND CONCLUSIONS.	98
16. REFERENCES	99

TABLE OF FIGURES

1. Plot of $F(X)$ against t	25
2. Interpretation of Iterative Scheme	26
3. Principle of steepest descent	58
4. Method of conjugate gradient	61

LIST OF TABLES

Table 1. The various components of \underline{X} at the end of each iteration. . . .	32
Table 2. Ratios of X_{n+1}^i/X_n^i for Seidel's process.	32
Table 3. Various iterates of "back-and-forth" Seidel process..	34
Table 4. Ratios of X_{n+1}^i/X_n^i for "back-and-forth" Seidel's process. . . .	35
Table 5. Favorable ranges of N for the Gauss elimination, linear iteration method and the Monte Carlo method.	53
Table 6. Results of various iterations by method of steepest descent. .	59
Table 7. Results of various iterations given by the conjugate gradient method.	63
Table 8. Comparison of the different solutions for the charged wire. . .	94
Table 9. Comparison of Gaussian elimination and conjugate gradient method for the solution vector $\underline{X} = [\underline{A}]^{-1}\underline{Y}$	96

1. INTRODUCTION

The problem of radiation and scattering from electromagnetic structures may be formulated in terms of the E-field, the H-field or the combined field integral equations. The integral equations are then reduced to matrix equations by the method of moments. Hence, the maximum size of an electromagnetic field problem that can be solved by this technique depends on how efficiently solutions of a set of simultaneous equations are obtained.

In system identification, on the other hand, the problem is formulated in terms of a convolution integral. When any of the standard techniques is utilized to identify the system, one again encounters a set of simultaneous equations. The only difference between the two cases is that in the former one often encounters a matrix which has large elements on the diagonal, whereas in the latter case the matrix may be nearly singular.

The objective of this report is to survey many of the popular methods for the solution of large matrix equations.

In section 2, a review is made of the direct methods for solving matrix equations. An analysis of round-off error is also made for these methods. In section 3, we present the philosophy of various iterative methods. In section 4, we discuss the various linear iterative methods and in section 5, the Monte Carlo methods. The Monte Carlo methods are statistical methods and are quite efficient in evaluating one component of the solution. Next, in section 6, comparison of efficiencies is made between Monte Carlo methods, Gaussian elimination, and linear iterative methods. In section 7, the various nonlinear iterative methods are discussed. The rates of convergence for the various methods are discussed in section 8. Section 9 presents the analysis of round-off errors associated with various iterative methods.

Section 10 extends the various methods to complex unsymmetric matrices. In section 11, we present a method for accelerating various iterative methods and reducing the round-off errors of direct methods. Sections 12 and 13 present the core storage and the work required for all the methods presented in this report. Section 14 presents a discussion on the conjugate gradient method.

Thus this presentation provides a comparison of the popular methods used to solve large systems of matrix equations.

In our discussion of the various methods, only references which are directly relevant are noted. No attempt has been made to cite the earliest sources. In many cases, additional references may be found in the papers mentioned.

2. DIRECT METHODS FOR SOLVING MATRIX EQUATIONS

Summary

In this section we present all the direct methods. These include Cramer's rule and the two versions of Gaussian elimination [LU decomposition and the compact method]. It is shown that the Gaussian elimination for the solution of $\underline{A} \underline{X} = \underline{Y}$ is optimum (insofar as the total number of operations is concerned) if one is restricted to handling one row or one column only of the matrix at a time for processing. The method due to Volker Strassen takes lesser computation than Gaussian elimination if one works with a block of the matrix at a time. Also it has been shown that Winograd's method of computing matrix products is much faster than the conventional way of multiplying matrices. Finally we show that the round-off error in direct methods is proportional to the condition number of \underline{A} . If ΔA and ΔY are the inaccuracies in the representation of \underline{A} and \underline{Y} , then the uncertainty ΔX in the solution \underline{X} is given by

$$\frac{\|\Delta \underline{X}\|_2}{\|\underline{X}\|_2} \leq \frac{\text{cond } [\underline{A}] \cdot \left\{ \frac{\|\Delta \underline{A}\|_2}{\|\underline{A}\|_2} + \frac{\|\Delta \underline{Y}\|_2}{\|\underline{Y}\|_2} \right\}}{1 - \sqrt{N} \cdot \text{cond } [\underline{A}] \cdot 2^{-t}} \leq \frac{2^{-t} [\sqrt{N}+1] \text{cond } [\underline{A}]}{1 - \sqrt{N} \text{cond } [\underline{A}] \cdot 2^{-t}}$$

where t is the number of binary digits with which computation is actually carried out in the computer and N is the dimension of \underline{A} .

In this section we describe exact methods for the numerical solution of systems of linear equations. By exact methods, we mean methods which give a solution of the problem by using a finite number of elementary arithmetic operations. If the initial elements of the matrix are given exactly and if the computations are carried out exactly, then the solution is also exact. In exact methods the number of computational operations necessary for solving a problem depends only on the type of computational scheme and on the order of the matrix which defines the problem.

2.1 CRAMER'S RULE (THE ADJOINT METHOD) [1]

This method is too well-known to elaborate and too cumbersome for practical use. Hence, only the final result is given. If

$$\underline{AX} = \underline{Y} \quad (2.1)$$

where A is a given NxN square matrix and X and Y are Nxl column matrices, then the unknown *i*th element of X is given by

$$x^i = |A|^{-1} \sum_j \{\text{adjoint } A\}^{ij} y^j \quad (2.2)$$

where $\{\text{adjoint } A\}^{ij}$ are the cofactors of the determinant of A denoted as $|A|$, and the superscripts represent the elements of the matrix. The solution of a system of *N* linear equations by use of Cramer's rule requires the evaluation of $(N+1)$ determinants of order *N*. If evaluated directly, each determinant requires $\alpha(N+1)!$ multiplications, where $1 \leq \alpha \leq 1.71828$.

Solution of all the unknown elements of X requires $\alpha(N+1)!$ multiplications, *N* divisions and $(N+1)!$ additions or subtractions. The other methods which we are going to discuss next require much less work.

2.2 GAUSSIAN ELIMINATION AND LU DECOMPOSITION [1,2,3]

Carl Frederick Gauss used this method to solve a system of linear algebraic equations. This method is based on the idea of eliminating the unknowns one at a time. A series of successive eliminations is carried out by which the given system $\underline{A}\underline{X} = \underline{Y}$ is transformed into a system with a triangular matrix, whose solution presents no difficulty. The factorization of \underline{A} as the product $\underline{L}\underline{U}$ is the basic idea of all Gaussian elimination schemes. Equivalently $\underline{A}\underline{X} = \underline{Y}$ can be rewritten as $\underline{L}\underline{U}\underline{X} = \underline{Y}$. Here \underline{L} is a lower triangular matrix (i.e. $L^{ij} = 0$ for $i < j$) and \underline{U} is an upper triangular matrix (i.e. $U^{ij} = 0$ for $i > j$). Thus $\underline{L}\underline{U}\underline{X} = \underline{Y}$ represents two triangular systems

$$\begin{aligned} \underline{L}\underline{G} &= \underline{Y} \\ \underline{U}\underline{X} &= \underline{G} \end{aligned} \tag{2.3}$$

which can be very easily solved. The calculation of \underline{L} and \underline{U} together with the solution of $\underline{L}\underline{G} = \underline{Y}$ is usually called the forward elimination and the solution of $\underline{U}\underline{X} = \underline{G}$ is the backward substitution. The computation of \underline{L} and \underline{U} is referred to as triangular decomposition. The various Gaussian elimination methods differ in the order in which computations are carried out in the forward elimination. Next we describe how the LU decomposition is carried out without pivoting.

Given the matrix \underline{A} and the vector \underline{Y} , we use elementary row operations to put zeros below the main diagonal of \underline{A} . Assume $A^{11} \neq 0$. [A^{ij} represent the element belonging to the i th row and j th column]. Let $M^{il} = \frac{A^{il}}{A^{11}}$. We then subtract M^{il} times the first equation from the i th equation, and also subtract M^{il} times Y^l from Y^i to obtain a set of equations which do not involve X^1 . This new set of $N-1$ equations along with the first equation of the original set can be written as

$$\underline{A}_2 \underline{X} = \underline{Y}_2 \quad (2.4)$$

where

$$\underline{A}_2 = \underline{M}_1 \underline{A}; \quad \underline{Y}_2 = \underline{M}_1 \underline{Y}, \text{ and}$$

$$\underline{M}_1 = \begin{bmatrix} 1 & & & & \\ -M^{21} & 1 & & & \\ -M^{31} & 0 & 1 & & \\ \dots & & & & \\ -M^{N1} & 0 & 0 & 1 & \\ & & & & 0 \end{bmatrix} \quad (2.5)$$

Next we assume $A_2^{22} \neq 0$. Let $M^{i2} = \frac{A^{i2}}{A^{22}}$. Then premultiply \underline{A}_2 and \underline{Y}_2 by \underline{M}_2 which is given by

$$\underline{M}_2 = \begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -M^{32} & 1 & & \\ 0 & -M^{42} & 0 & 1 & \\ \dots & & & & \\ 0 & -M^{N2} & 0 & \dots & 1 \end{bmatrix}$$

Thus $\underline{A}_3 = \underline{M}_2 \underline{A}_2$ and $\underline{Y}_3 = \underline{M}_2 \underline{Y}_2$. This corresponds to eliminating x^2 from the last $N-2$ equations. We continue the process until we obtain the following structure

$$\underline{U}\underline{X} = \underline{G}$$

or

$$\begin{bmatrix} A^{11} & A^{12} & A^{13} & \dots & A^{1N} \\ A_2^{22} & A_2^{23} & \dots & A_2^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & & & A_N^{NN} \end{bmatrix} \times \begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ X^N \end{bmatrix} = \begin{bmatrix} Y^1 \\ Y^2 \\ \vdots \\ Y_N^N \end{bmatrix} \quad (2.7)$$

Let $\underline{M} = M_N \ M_{N-1} \ \dots \ M_1$. Then since $\underline{M}\underline{A} = \underline{U}$ we have $\underline{A} = \{\underline{M}\}^{-1} \underline{U}$. Thus $\{\underline{M}\}^{-1} = \{M_1\}^{-1} \times \{M_2\}^{-1} \times \dots \times \{M_N\}^{-1}$. We now define $\underline{L} = \{\underline{M}\}^{-1}$ and obtain

$$\underline{L} = \{\underline{M}\}^{-1} = \begin{bmatrix} 1 & & & & \\ M^{21} & 1 & & & \\ M^{31} & M^{32} & 1 & & \\ \dots & & & \ddots & \\ M^{N1} & M^{N2} & \dots & \dots & 1 \end{bmatrix} \quad (2.8)$$

Observe $\underline{A} = \{\underline{M}\}^{-1} \underline{U} = \underline{LU}$. Note that the matrix \underline{M} is never actually formed.

As the elimination progresses, the below diagonal elements M^{ij} of \underline{L} are stored in place of the below diagonal elements of \underline{A} , and the elements U^{ij} of \underline{U} are stored in place of the diagonal and above diagonal elements of \underline{A} .

At the end we have

$$\begin{bmatrix} U^{11} & U^{12} & U^{13} & \dots & U^{1N} \\ M^{21} & U^{22} & U^{23} & \dots & U^{2N} \\ M^{31} & M^{32} & U^{33} & \dots & U^{3N} \\ \dots & & & & \\ M^{N1} & M^{N2} & M^{N3} & \dots & U^{NN} \end{bmatrix} \quad (2.9)$$

stored in place of \underline{A} . Triangular decomposition is thus summarized by the facts that \underline{L} is simply the matrix of multipliers M^{ij} with a diagonal of 1's and the \underline{U} is the matrix A_N of (2.7). Also note that the intermediate solution \underline{G} of (2.3) is \underline{Y}_N . The processing of \underline{Y} , i.e. the transformation of \underline{Y} into \underline{Y}_N , can be done simultaneously with the processing of \underline{A} . Since we have all the necessary multipliers stored however, it can just as well be done at the end.

This describes ordinary Gaussian elimination without pivoting. The term "pivoting" is used to describe row and column interchanges at the k th stage of elimination to move the largest element in absolute value in the remaining unchanged $(N-k+1) \times (N-k+1)$ matrix to the k th diagonal element. Thus the pivot at the k th stage (or the diagonal element A_k^{kk}) is chosen as the element of largest absolute value in the submatrix A_k composed of columns k through N and rows k through N . Hence, both row and column interchanges are necessary to bring the pivot to the k th diagonal position. Use of pivots has two advantages. First, it relaxes the assumption that $A_k^{kk} \neq 0$ and secondly, the use of a pivoting strategy reduces the round-off error of the LU decomposition process. The analysis of round-off error for this process is presented in section 2.5. Also note that pivoting requires more operations.

2.3 COMPACT METHOD (CROUT AND DOLITTLE OR CHOLESKY'S METHOD) [1,2,3]:

This method depends explicitly on the triangular resolution of \underline{A} as \underline{LU} , that is, the elements of \underline{L} and \underline{U} are all computed and used. It is termed a compact scheme since the elements in the final triangular form are obtained by accumulation, dispensing with the computation and recording of the intermediate A_k^{ij} elements and thereby reducing round-off error.

Since $\underline{A} = \underline{LU}$, the equation for the elements of \underline{L} and \underline{U} is

$$\sum_{k=1}^{\min(i,j)} L^{ik} U^{kj} = A^{ij}$$

Letting $L^{kk} = 1$, for $k = 1, 2, \dots, N$, we have N^2 equations in N^2 unknowns.

$$L^{kk} = 1$$

$$U^{kj} = A^{kj} - \sum_{m=1}^{k-1} L^{km} U^{mj} \text{ for } j = k, \dots, N$$

$$L^{ik} = \frac{1}{U^{kk}} [A^{ik} - \sum_{m=1}^{k-1} L^{im} U^{mk}] \text{ for } i = k+1, \dots, N$$

$$L^{ik} = 0 \text{ for } i < k$$

$$U^{kj} = 0 \text{ for } j < k \quad (2.10)$$

Hence, the order of elimination is first row of U , first column of L , second row of U , second column of L and so on. As the elements L^{ik} and U^{kj} are computed they are written over A in the obvious way. After obtaining elements of L and U , we solve $AX = Y$ by writing $LUX = Y$ which is then equivalent to solving the triangular systems $UX = G$ and $LG = Y$.

The accuracy of the method can be improved if pivoting is introduced. After the row U^{ik} , $i=k, \dots, N$ is computed, the largest U^{ik} in absolute value, say U^{jk} may be selected as U^{kk} , and its column the (jth) interchanged with the kth column in both U and A . This should not cause any problem even when L and U are written over A . Then the next row of L , L^{kj} , for $j = K+1, \dots, N$ is computed. Whenever a new row of U is computed, the largest of its elements in absolute value is chosen as diagonal. The L and U matrices so obtained are not the triangular decomposition of A but are the decomposition of A , where

of \tilde{A} , where

$\tilde{A} = (I^{i_N} \dots I^{i_2^2} I^{i_1^1}) A$, obtained from A by a sequence of row interchanges, I^{i_k} , of the i_k row with the k th row, where $k=1, \dots, N$.

When the matrix A is symmetric, this method is often referred to as the square-root method.

2.4 IS GAUSSIAN ELIMINATION REALLY OPTIMUM? [2]

Gaussian elimination, as presented in the previous section, is really optimum if and only if one is interested in handling the elements of the matrices by rows or by columns. Under those conditions Klyuyev and Kokovkin-Scherbak [4] have proved that no general system of linear equations can be solved with fewer arithmetic operations than are required by Gaussian elimination. If $\Theta(N^2)$ is defined as the terms of the order of N^2 , then in general Gaussian elimination requires $\frac{N^3}{3} + \Theta(N^2)$ multiplications and $\frac{N^3}{3} + \Theta(N^2)$ additions. However, Winograd [5,6] has shown that a general system of linear equations can be solved in $\frac{N^3}{6} + \Theta(N^2)$ multiplications and $\frac{N^3}{2} + \Theta(N^2)$ additions. Since multiplications require more time than additions, Winograd's method would be faster than Gaussian elimination. It is interesting to observe that the total number of multiplications and additions for both Gaussian elimination and Winograd's method is $\frac{2}{3}N^3 + \Theta(N^2)$. Recently Volker Strassen [7] has shown that it is possible to solve a general system of linear equations with $\Theta(N^p)$ arithmetic operations, where in this case $p = \log_2 7 = 2.807$. However, it is not known whether this value of p is the minimum exponent.

2.5 ANALYSIS OF ROUND-OFF ERRORS FOR DIRECT METHODS OF SOLVING A SYSTEM OF LINEAR EQUATIONS [2,8]

The solution of a set of equations by Gaussian elimination is based on the triangularization of a matrix. If we start with

$$\underline{A}\underline{X} = \underline{Y} \quad \text{or} \quad \underline{A}_1\underline{X} = \underline{Y}_1$$

in Gaussian elimination, then the following (N-1) equivalent sets are produced.

$$\underline{A}_r \underline{X} = \underline{Y}_r \quad \text{for } r = 2, 3, \dots, N \quad (2.11)$$

The matrix \underline{A}_N of the final set is of upper triangular form. In general \underline{A}_r is of triangular form as regards to its first r rows, and it has a square matrix of non-zero elements in the bottom right hand corner. The square matrix is of order $N+1 - r$. The matrix \underline{A}_{r+1} is derived from \underline{A}_r by subtracting a multiple M^{ir} of the r th row from the i th row for values of i from $r+1$ to N . The multipliers M^{ir} are defined by

$$M^{ir} = \frac{A_r^{ir}}{A_r^{rr}} \quad (2.12)$$

The r th row of \underline{A}_r is called the r th pivoted row and A_r^{rr} is called the r th pivot.

In order to obtain the elements A_r^{ij} for $i \leq j$, we write

$$A_2^{ij} = A_1^{ij} - M^{i1} A_1^{1j} + \epsilon_2^{ij}$$

$$A_3^{ij} = A_2^{ij} - M^{i2} A_2^{2j} + \epsilon_3^{ij}$$

.....

$$A_r^{ij} = A_{r-1}^{ij} - M^{i, r-1} A_{r-1}^{r-1, j} + \epsilon_r^{ij} \quad (2.13)$$

where all A_r^{ij} and M^{ir} refer to computed values and ε_r^{ij} is the difference between the exact A_r^{ij} and the value which is obtained using the computed

A_{r-1}^{ij} , $M^{i,r-1}$ and A_{r-1}^{r-1}, j . After summing the equations in (2.13) we obtain

$$A_r^{ij} = A_1^{ij} - M^{i1} A_1^{1j} - M^{i2} A_2^{2j} - \dots - M^{i,r-1} A_{r-1}^{r-1, j} + e^{ij} \quad (2.14)$$

where $e^{ij} = \varepsilon_2^{ij} + \varepsilon_3^{ij} + \dots + \varepsilon_r^{ij}$ (2.15)

For $i > j$ the elements of A_r are modified until A_j^{ij} is obtained. A_j^{ij} is then used to compute M^{ij} , and A_{j+1}^{ij} to A_N^{ij} are all taken to be exactly equal to zero. The equations are therefore

$$\begin{aligned} A_2^{ij} &= A_1^{ij} - M^{i1} A_1^{1j} + \varepsilon_2^{ij} \\ A_3^{ij} &= A_2^{ij} - M^{i2} A_2^{2j} + \varepsilon_3^{ij} \\ &\dots \\ A_j^{ij} &= A_{j-1}^{ij} - M^{i,j-1} A_{j-1}^{j-1, j} + \varepsilon_j^{ij} \\ 0 &= A_j^{ij} - M^{ij} A_j^{jj} + \varepsilon_{j+1}^{ij} \end{aligned} \quad (2.16)$$

Again summing all the equations in (2.16) we have

$$0 = A_1^{ij} - M^{i1} A_1^{1j} - M^{i2} A_2^{2j} - \dots - M^{ij} A_j^{jj} + e^{ij} \quad (2.17)$$

where $e^{ij} = \varepsilon_2^{ij} + \varepsilon_3^{ij} + \dots + \varepsilon_{j+1}^{ij}$ (2.18)

If we take the terms involving M^{ij} to the left hand sides in equations (2.14) and (2.17) the set of N^2 equations then reduce to the single matrix equation

$$\underline{L}\underline{U} = \underline{A}_1 + \underline{E} = \underline{A} + \underline{E} \quad (2.19)$$

where \underline{L} is the lower triangular matrix defined by

$$\underline{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ M^{21} & 1 & 0 & \dots & \dots & 0 \\ M^{31} & M^{32} & 1 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ M^{N1} & M^{N2} & M^{N3} & \dots & \dots & 1 \end{bmatrix} \quad (2.20)$$

and U is the upper triangular matrix defined as

$$\underline{U} = \begin{bmatrix} A_1^{11} & A_1^{12} & \dots & \dots & \dots & A_1^{1N} \\ 0 & A_2^{22} & \dots & \dots & \dots & A_2^{2N} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & A_N^{NN} \end{bmatrix} \quad (2.21)$$

and E is the error matrix defined by (2.15) for $i \leq j$ and by (2.18) for $i > j$.

In actual computation, we select pivotal rows so as to ensure

$$|M^{ij}| \leq 1 \quad (2.22)$$

There are two main ways this is done. In the first case the columns may be eliminated in the natural order, but at the r th stage, the pivotal row is taken to be that one of the remaining $N+1 - r$ rows which has the largest element in magnitude in column r . This is called partial pivoting.

Secondly, if at each r th stage we select the largest element in magnitude from the remaining $N+1 - r$ rows, this is called complete pivoting. Wilkinson has shown that [8]

$$|A_r^{ij}| \leq 2^{r-1} a \text{ (for partial pivoting)} \quad (2.23)$$

and

$$|A_r^{rr}| \leq r^{\frac{1}{2}} \{2^1 \cdot 3^{\frac{1}{2}} \cdot 4^{1/3} \cdot \dots \cdot r^{\frac{1}{r-1}}\}^{\frac{1}{r-1}} a \text{ (for complete pivoting)}$$

where a is given by

$$| A_{ij}^{ij} | \triangleq | A_1^{ij} | \leq a \quad (2.25)$$

Wilkinson claims that for almost all matrices A ,

$$| A_r^{rr} | \leq ra \quad (2.26)$$

We denote

$$\max | A_r^{ij} | = g \quad (2.27)$$

$$\text{and } \| A_r^{ij} \| \triangleq \max_i \sum_{j=1}^N | A_r^{ij} | \leq Ng \quad (2.28)$$

Now we try to find $\| E \|$ - the total error encountered in the triangular decomposition of \underline{A} . We observe

$$A_r^{ij} = \{ A_{r-1}^{ii} - M^{i,r-1} A_{r-1}^{r-1,j} \} (1+\varepsilon_2) \quad (2.29)$$

where ε_1 and ε_2 are the round-off errors made in the multiplication and subtraction, respectively. We know

$$| \varepsilon_1 | = | \varepsilon_2 | \leq 2^{-t} \text{ (in binary)} \quad (2.30)$$

$$\leq \frac{1}{2} 10^{1-t} \text{ (in decimal)} \quad (2.31)$$

where t is the number of digits in which actual computation is carried out.

Thus

$$\begin{aligned} \varepsilon_r^{ij} &= A_r^{ij} - \left\{ \frac{A_r^{ij}}{1 + \varepsilon_2} + M^{i,r-1} A_{r-1}^{r-1,j} \varepsilon_1 \right\} = \frac{A_r^{ij} \varepsilon_2}{1 + \varepsilon_2} - M^{i,r-1} A_{r-1}^{r-1,j} \varepsilon_1 \\ &\leq g 2^{-t} \left\{ \frac{1}{1 - 2^{-t}} + 1 \right\} = \frac{2^{t+1}-1}{2^t - 1} \cdot g \cdot 2^{-t} \end{aligned} \quad (2.32)$$

This applies to ε_r^{ij} except ε_{j+1}^{ij} for $i > j$. For these we find

$$| M^{jj} | \leq 1 \text{ (from 2.22)}$$

$$| A_r^{jj} | \leq g \text{ (from 2.27)}$$

$$|\epsilon_{j+1}^{ij}| = \left| A_j^{ij} \cdot \frac{A_{jj}^{jj} \epsilon_1}{A_{jj}^{jj}} \right| \leq g \cdot 2^{-t} < \frac{2^{t+1}-1}{2^t-1} \cdot g \cdot 2^{-t} \quad (2.33)$$

so that we need not give these elements special treatments. Combining (2.15), (2.18), (2.32), and (2.33) we have for complete pivoting

$$|E^{ij}| \leq \frac{2^{t+1}-1}{2^t-1} \cdot g \cdot 2^{-t} \quad \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 2 & 2 & \dots & 2 & 2 \\ 1 & 2 & 3 & \dots & 3 & 3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 2 & 3 & \dots & (N-1) & (N-1) \end{bmatrix} \quad (2.34)$$

$$\text{Thus } ||E|| \triangleq \max_i \sum_{j=1}^N |E^{ij}| \leq \frac{2^{t+1}-1}{2^t-1} \cdot g \cdot 2^{-t} \cdot (\frac{N}{2} + 1) \cdot (N-1) \quad (2.35)$$

So in summary, what we have done so far is expressed $\underline{AX} = \underline{Y}$ in the form $\underline{LUX} = \underline{Y}$. The computed \underline{L} and \underline{U} satisfy $\underline{LU} = \underline{A} + \underline{E}$ and, hence, if we solve $\underline{LUX} = \underline{Y}$ without any further rounding error, we would obtain

$$(\underline{A} + \underline{E}) \underline{X} = \underline{Y} \quad (2.36)$$

Gaussian elimination solves (2.36) in two steps

$$\underline{LG} = \underline{Y} \quad (2.37)$$

$$\underline{UX} = \underline{G} \quad (2.38)$$

each of which requires the solution of a set of equations having a triangular matrix of coefficients. We therefore now consider the errors made in solving triangular matrix equations. From (2.37) we find [8]

$$G^r = \frac{-L^{r1} G^1 (1 + \theta^{r1}) - L^{r2} G^2 (1 + \theta^{r2}) - \dots - L^{r,r-1} G^{r-1} (1 + \theta^{r,r-1}) + Y^r (1 + \varepsilon^r)}{L^{rr} (1 + \delta^r)} \quad (2.39)$$

where θ^{ri} , ε^r and δ^r are the round off errors associated with L^{ri} , Y^r and

L^{rr} respectively. we have from [8]

$$|\varepsilon^r| \leq 2^{-t} \quad (2.40)$$

$$|\delta^r| \leq 2^{-t} \quad (2.41)$$

$$\text{and } |\phi^{ri}| \leq (r+2-i) 2^{-t} \quad (2.42)$$

Rewriting (2.39) we find

$$G^r = \frac{-L^{r1} G^1 \frac{1 + \theta^{r1}}{1 + \varepsilon^r} - L^{r2} G^2 \frac{1 + \theta^{r2}}{1 + \varepsilon^r} - \dots - L^{r, r-1} G^{r-1} \frac{1 + \theta^{r, r-1}}{1 + \varepsilon^r} + \gamma^r}{L^{rr} \frac{1 + \delta^r}{1 + \varepsilon^r}} \quad (2.43)$$

$$\text{and if } \frac{1 + \theta^{ri}}{1 + \varepsilon^r} = 1 + \psi^{ri} \quad (2.44)$$

$$\text{and } \frac{1 + \delta^r}{1 + \varepsilon^r} = 1 + \psi^{rr} \quad (2.45)$$

then (2.43) can be expressed as

$$\sum_{i=1}^r L^{ri} G^i (1 + \psi^{ri}) = \gamma^r \quad (2.46)$$

$$\text{Since } -2^{-t} \leq \delta^r \leq 2^{-t} \text{ and } -2^{-t} \leq \varepsilon^r \leq 2^{-t} \text{ we have } |\psi^{rr}| \leq 2 \times 2^{-t} \quad (2.47)$$

and from (2.42) and (2.44)

$$(r+2-i) 2^{-t} \geq |\phi^{ri}| \geq |\psi^{ri}| + |\varepsilon^r| \quad (2.48)$$

or $(r+1-i) 2^{-t} \geq |\psi^{ri}|$

Equations (2.43), (2.47) and (2.48) show that the computed vector is the exact solution of

$$(\underline{L} + \Delta \underline{L}) \underline{u} = \underline{\gamma}$$

where $\Delta \underline{L}$ is bounded by [8]

$$|\underline{\Delta L}| \leq 2^{-t} \times \begin{bmatrix} 2^1 |L^{11}| \\ 2^2 |L^{21}| & 2^2 |L^{22}| \\ 2^3 |L^{31}| & 2^3 |L^{32}| & 2^3 |L^{33}| \\ \dots & \dots & \dots \\ N |L^{N1}| & (N-1) |L^{N2}| & \dots & \dots & 2^{N-1} |L^{NN}| \end{bmatrix} \quad (2.49)$$

Hence [8]

$$||\underline{\Delta L}|| \triangleq \max_i \sum_{j=1}^N |\underline{\Delta L}^{ij}| \leq \frac{1}{2} (N^2 + N + 2) 2^{-t} \quad (2.50)$$

$$\text{as } \max_{i,j} |\underline{\Delta L}^{ij}| < 1$$

Similarly we can prove [8]

$$||\underline{\Delta U}|| \leq \frac{1}{2} (N^2 + N + 2) \leq 2^{-t} \quad (2.51)$$

Also we have [8]

$$||\underline{L}|| \triangleq \max_i \sum_{j=1}^N |L^{ij}| \leq N \quad (2.52)$$

and [8]

$$||\underline{U}|| \triangleq \max_i \sum_{j=1}^N |U^{ij}| \leq Ng \quad (2.53)$$

Hence X satisfies

$$(\underline{L} + \underline{\Delta L}) (\underline{U} + \underline{\Delta U}) (\underline{X} + \underline{\Delta X}) = (\underline{Y} + \underline{\Delta Y}) \quad (2.54)$$

where ΔX is the uncertainty in the solution due to the uncertainties associated with the computing processes. Thus if E is the error associated with the representation of A, (i.e. A + E = LU), then

$$(\underline{A} + \underline{E} + \underline{L} \cdot \underline{\Delta U} + \underline{\Delta L} \cdot \underline{U} + \underline{\Delta L} \cdot \underline{\Delta U}) (\underline{X} + \underline{\Delta X}) = (\underline{Y} + \underline{\Delta Y})$$

$$\text{or } (\underline{A} + \underline{\Delta A}) (\underline{X} + \underline{\Delta X}) = (\underline{Y} + \underline{\Delta Y}) \quad (2.55)$$

where $\underline{\Delta A} = \underline{E} + \underline{L} \cdot \underline{\Delta U} + \underline{\Delta L} \cdot \underline{U} + \underline{\Delta L} \cdot \underline{\Delta U}$

and so from (2.35), (2.50) - (2.53) we obtain

$$\begin{aligned} \|\underline{\Delta A}\| &\leq g 2^{-t} \left[\frac{N^4 \cdot 2^{-t}}{4} + N^3 (2^{-t-1} + 1) + N^2 \left(\frac{5}{4} 2^{-t} + 1 + \frac{2^t - 0.5}{2^t - 1} \right) + \dots \right] \\ &\leq g 2^{-t} \left[N^3 + 2N^2 + \frac{N^4 2^{-t}}{4} + \dots \right] \end{aligned} \quad (2.56)$$

So far we have discussed only the infinity norm, i.e.

$$\|\underline{A}\|_{\infty} \triangleq \max_i \sum_{j=1}^N |A^{ij}|$$

We next introduce the Euclidean norm. The Euclidean norm is defined as

$$\|\underline{A}\|_E \triangleq \left\{ \sum_{i=1}^N \sum_{j=1}^N |A^{ij}|^2 \right\}^{\frac{1}{2}} \quad (2.57)$$

It can be shown that under the Euclidean norm [8]

$$\|\underline{E}\|_E \leq \frac{2^{t+1}-1}{2^t-1} g 2^{-t} N \sqrt{\frac{N^2-1}{6}} \quad (2.58)$$

$$\|\underline{L}\|_E \leq \sqrt{\frac{N(N+1)}{2}} \quad (2.59)$$

$$\|\underline{U}\|_E \leq 8 \sqrt{\frac{N(N+1)}{2}} \quad (2.60)$$

$$\|\underline{\Delta L}\|_E \leq \frac{(N+2)^2 2^{-t}}{\sqrt{12}} \quad (2.61)$$

$$\|\underline{\Delta U}\|_E \leq \frac{(N+2)^2 g 2^{-t}}{\sqrt{12}} \quad (2.62)$$

$$\|\underline{\Delta A}\|_E \leq g 2^{-t} \left\{ \frac{N^4 2^{-t}}{12} + \frac{N^3}{\sqrt{6}} + 3N^2 + \dots \right\} \quad (2.63)$$

However, Wilkinson claims that for all practical purposes [8]

$$\|\underline{\Delta A}\|_E \leq N \cdot g \cdot 2^{-t} \quad \text{and} \quad \|\underline{\Delta A}\|_{\infty} \leq N \cdot g \cdot 2^{-t}$$

So for both the Euclidean and infinity norms

$$\frac{\|\underline{\Delta A}\|}{\|\underline{A}\|} \leq \frac{2^{-t} \cdot N \cdot g}{N \cdot g} \leq 2^{-t} \quad (2.64)$$

Since $\|\underline{A}\|_E$ and $\|\underline{A}\|_{\infty}$ are not related to the eigenvalues of the matrix \underline{A} ,

we introduce the spectral norm. It is defined as

$$\|\underline{A}\|_2 \triangleq |\text{maximum eigenvalue of } \underline{A}| \triangleq \lambda_{\max}[\underline{A}] \quad (2.65)$$

and cond $[\underline{A}] \triangleq \text{condition number of } \underline{A} \triangleq \|\underline{A}^{-1}\|_2 \cdot \|\underline{A}\|_2 \triangleq \frac{\lambda_{\max}[\underline{A}]}{\lambda_{\min}[\underline{A}]}$

$$(2.66)$$

For the spectral norm it can be shown that [8]

$$\|\underline{A}\|_2 \leq \|\underline{A}\|_E \leq \sqrt{N} \|\underline{A}\|_2 \quad (2.67)$$

Thus,

$$\|\underline{\Delta A}\|_2 \leq \|\underline{\Delta A}\|_E \leq 2^{-t} \|\underline{A}\|_E \leq \sqrt{N} 2^{-t} \|\underline{A}\|_2, \quad (2.68)$$

and

$$\frac{\|\underline{\Delta A}\|_2}{\|\underline{A}\|_2} \leq \sqrt{N} \cdot 2^{-t} \quad (2.69)$$

However, note that for a row or column matrix [8]

$$\|\underline{Y}\|_2 \equiv \|\underline{Y}\|_E \triangleq \sqrt{(Y^1)^2 + (Y^2)^2 + \dots}$$

In order to find $\underline{\Delta X}$ from (2.55) we obtain

$$\underline{\Delta X} = \{\underline{I} + [\underline{A}]^{-1} \cdot \underline{\Delta A}\}^{-1} \cdot \{[\underline{A}]^{-1} \cdot \underline{\Delta Y} - [\underline{A}]^{-1} \cdot \underline{\Delta A} \cdot \underline{X}\} \quad (2.70)$$

where \underline{I} is the identity matrix. If $\|\underline{A}^{-1} \cdot \underline{\Delta A}\|_2 < 1$ then it can be shown

[8, p. 92]

$$\|\underline{I} + [\underline{A}]^{-1} \cdot \underline{\Delta A}\|_2 \leq \frac{1}{1 - \|\underline{A}^{-1}\|_2 \cdot \|\underline{\Delta A}\|_2} \quad (2.71)$$

Then we have

$$\begin{aligned} \frac{\|\underline{\Delta X}\|_2}{\|\underline{X}\|_2} &\leq \frac{\|\underline{A}^{-1}\|_2 \cdot \|\underline{A}\|_2}{1 - \|\underline{A}^{-1}\|_2 \cdot \|\underline{A}\|_2 \left\{ \frac{\|\underline{\Delta A}\|_2}{\|\underline{A}\|_2} \right\}} \left\{ \frac{\|\underline{\Delta Y}\|_2}{\|\underline{Y}\|_2} + \frac{\|\underline{\Delta A}\|_2}{\|\underline{A}\|_2} \right\} \\ &\leq \frac{\text{cond } [\underline{A}]}{1 - \sqrt{N} \cdot \text{cond } [\underline{A}] \cdot 2^{-t}} [2^{-t} + \sqrt{N} \cdot 2^{-t}] \\ &\leq \frac{2^{-t} [\sqrt{N} + 1] \text{cond } [\underline{A}]}{1 - \sqrt{N} \cdot 2^{-t} \text{cond } [\underline{A}]} \end{aligned} \quad (2.72)$$

Thus there is absolutely no way to recognize an accurate solution \underline{X} given by Gaussian elimination unless

$$\sqrt{N} \cdot 2^{-t} \cdot \text{cond } [\underline{A}] \ll 1 \quad (2.73)$$

Thus (2.72) relates the accuracy of the solution to the dimension of \underline{A} , its condition number, and the number of digits with which computation has been carried out.

As an example consider the solution of the following problem $\underline{A}\underline{x} = \underline{y}$ by Gaussian elimination. Let \underline{A} be the ill-conditioned Hilbert matrix and \underline{y} be that vector for which $\underline{x} = \{1, 2, 3, \dots, N\}$. The problem then is to find \underline{x} given \underline{A} and \underline{y} . We simulated this problem on the Xerox Sigma-9 computer where computation is carried out using twenty four binary digits.

For a 4th order Hilbert matrix the condition number is obtained as

$\text{cond } (\underline{A}_4) = 1.55 \times 10^4$ [2]. Thus for a 4th order Hilbert matrix (2.72)

reduces to $\frac{\|\underline{\Delta x}\|}{\|\underline{x}\|} \leq .00278$

and hence a very good accuracy in the results is expected. However for a fifth order Hilbert matrix $\frac{\|\underline{\Delta x}\|}{\|\underline{x}\|} \leq .09825$

since $\text{cond } (\underline{A}_5) = 4.77 \times 10^5$ [2]. This is reflected in the following results:

$$\begin{array}{cccccc} \underline{x} & \rightarrow & 0.996 & 2.067 & 2.708 & 4.442 & 4.783 \\ \frac{\|\underline{\Delta x}\|}{\|\underline{x}\|} & \rightarrow & .07772 & & & & \end{array}$$

Note that the theoretical error bound is large. For a sixth order Hilbert matrix we have

$$\begin{array}{ccccccc} \underline{x} & \rightarrow & 1.007 & 1.786 & 4.509 & -.0342 & 9.543 & 4.182 \\ \underline{x} \text{ exact} & \rightarrow & 1.0 & 2.0 & 3.0 & 4.0 & 5.0 & 6.0 \end{array}$$

In this case $\frac{\|\underline{\Delta x}\|}{\|\underline{x}\|} = -2.59$ (from 2.72)

since $\text{cond } (\underline{A}_6) = 1.5 \times 10^7$ from [2].

These results prompt us to look for alternate methods in which we could reduce the effects of round-off error in solving a system of equations. The effect of round-off error may become pronounced not only for very ill-conditioned matrices but also for large systems of equations in which a large number of arithmetic operations must be carried out. Iterative methods are good alternatives to rectify this problem of round-off error. For example, for a 7th order Hilbert matrix, where we know direct methods would not work, we obtained this result by the

conjugate gradient method at the end of seven iterations.

$$\begin{array}{ccccccc} \underline{x} & \rightarrow .993; & 2.090; & 2.678; & 4.034; & 5.208; & 6.121; & 6.780 \\ \underline{x}_{\text{exact}} & \rightarrow 1.; & 2.; & 3.; & 4.; & 5.; & 6.; & 7.; \end{array}$$

2.6 CONCLUSIONS

The direct methods are quite efficient when we have a well-conditioned matrix of small rank N . However, if the matrix A is ill-conditioned, then the direct methods may fail even for a 5×5 matrix. Also if the rank of the matrix is very large, then the round-off error may build up to make $\sqrt{N} \cdot \text{cond}(A) \cdot 2^{-t}$ comparable to unity. So we must look for alternative methods of solving systems of linear equations when we have a large matrix or a very ill-conditioned matrix. Even though the two conditions under which the direct methods fail are quite different, the disease is the same - round-off error. Thus we next look into the iterative methods, where the unknown is refined at each stage until we get the exact solution. In iterative methods, the round-off error is generally limited to the last iteration only. This we demonstrated by solving a 7×7 Hilbert matrix using single precision computation. We know that for this problem direct methods would fail as $\frac{\|\Delta x\|}{\|x\|} = -1.40$ (from 2.72).

$$\frac{\|\Delta x\|}{\|x\|}$$

3. PHILOSOPHY OF ITERATIVE METHODS FOR SOLVING MATRIX EQUATIONS

Summary

The basic philosophy of the iterative methods is discussed in this section. It is shown that the solution of the set of equations $\underline{A} \underline{X} = \underline{Y}$ is equivalent to the maximization/minimization of the functional $F(\underline{X}) = \frac{1}{2} \langle \underline{AX}, \underline{X} \rangle - \langle \underline{Y}, \underline{X} \rangle$ if \underline{A} is negative/positive definite. The contours of constant $F(\underline{X})$ are generally N-dimensional ellipsoids. Also the residuals $\underline{R}_n (= \underline{AX}_n - \underline{Y})$ at the end of each step are normals to the ellipsoid at \underline{X}_n . The paths \underline{P} by which one reaches the center point of the ellipsoid (which is the solution, $\underline{X}_{\text{exact}}$) are different for the different iterative methods. An iterative process is called linear if the present estimate \underline{X}_n is a linear combination of the past estimates $\underline{X}_0, \underline{X}_1, \underline{X}_2, \dots, \underline{X}_{n-1}$. Otherwise the iterative process is nonlinear. A process is called a stationary iterative process if the rule by which \underline{X}_n is determined does not change from iteration to iteration. Otherwise the iterative process is called nonstationary. Nonstationary methods are not pursued in this presentation because some ideas are needed about the magnitudes of the maximum and the minimum eigenvalues of \underline{A} for these methods to be effective.

3.1 MATHEMATICAL APPROACH OF ITERATIVE METHODS [1, 9, 10, 11]

Many boundary value problems of mathematical physics may be reduced to the solution of a matrix equation

$$\underline{AX} = \underline{Y} \quad (3.1)$$

The iterative method consists of choosing a trial function \underline{X}_0 for \underline{X} in (3.1). For this trial vector we have a residual \underline{R}_0 given by

$$\underline{R}_0 = \underline{AX}_0 - \underline{Y} \quad (3.2)$$

The objective of any iterative scheme is to alter the vector \underline{X}_0 systematically in such a way that the residuals eventually disappear. To achieve our goal we introduce the quadratic functional $F(\underline{X})$ defined as [9].

$$F(\underline{X}) = \frac{1}{2} \langle \underline{AX}, \underline{X} \rangle - \langle \underline{Y}, \underline{X} \rangle \quad (3.3)$$

Here $\langle \cdot, \cdot \rangle$ is the usual definition of the inner product. [In the present chapters we will assume A to be symmetric and A, X, Y are real matrices. We will derive the rates of convergence of various iterative schemes based on this assumption. Later we will extend the discussions to complex matrices by changing the definition of the inner product.] If we want to minimize or maximize the quadratic functional $F(X)$ defined by (3.3) then the first functional derivative should be made equal to zero. [This functional derivative is often referred to as the Frechet differential of $F(X)$].

The first differential is obtained as

$$F'(\underline{X}) = \langle \underline{AX} - \underline{Y}, \underline{\Delta X} \rangle = \langle \underline{R}, \underline{\Delta X} \rangle \quad (3.4)$$

The second differential of $F''(\underline{X})$ is obtained as

$$F''(\underline{X}) = \langle \underline{A\Delta X}, \underline{\Delta X} \rangle \quad (3.5)$$

Thus the solution of a symmetric system of matrix equations in (3.1) is equivalent to the problem of finding the minimum/maximum of a quadratic functional $F(\underline{X})$ of (3.3) depending on whether A is positive/negative definite.

In order to maximize/minimize $F(\underline{X})$ we start with a trial vector \underline{x}_0 . We select some direction \underline{P}_0 and correct \underline{x}_0 in the direction of \underline{P}_0 with the intention of approaching the maximum/minimum of $F(\underline{X})$. From now on let us assume A is positive definite so that we can explain the principles of an iterative scheme. The new trial vector \underline{x}_1 obtained at the end of the first iterative step is related to \underline{x}_0 by

$$\underline{x}_1 = \underline{x}_0 + t \underline{P}_0 \quad (3.6)$$

where t is a scalar parameter. Then

$$\begin{aligned} F(\underline{x}_1) &= F(\underline{x}_0 + t \underline{P}_0) \\ &= \frac{t^2}{2} \langle \underline{A}\underline{P}_0, \underline{P}_0 \rangle + t \langle \underline{A}\underline{x}_0 - \underline{Y}, \underline{P}_0 \rangle + F(\underline{x}_0) \end{aligned} \quad (3.7)$$

[NOTE: x_n^k represents the k th element of X obtained after n iterations.]

The parameter t is now selected in such a way $F(\underline{x}_1)$ reaches a minimum (as A has been assumed to be positive definite), i.e.

$$\begin{aligned} \frac{dF(\underline{x}_1)}{dt} &= t \langle \underline{A}\underline{P}_0, \underline{P}_0 \rangle + \langle \underline{A}\underline{x}_0 - \underline{Y}, \underline{P}_0 \rangle = t \langle \underline{A}\underline{P}_0, \underline{P}_0 \rangle + \langle \underline{R}_0 \underline{P}_0 \rangle \\ &= 0 \end{aligned}$$

$$\text{or } t_{\min} = -\frac{\langle \underline{R}_0, \underline{P}_0 \rangle}{\langle \underline{A}\underline{P}_0, \underline{P}_0 \rangle} \quad (3.8)$$

The second derivative of $F(\underline{x}_1)$ with respect to t yields

$$\frac{d^2 F(\underline{x}_1)}{dt^2} = \langle \underline{A}\underline{P}_0, \underline{P}_0 \rangle > 0 \quad (3.9)$$

since A is positive definite.

When A is nonsingular but indefinite, its solution makes the corresponding quadratic function stationary but not maximum/minimum.

In the event A is positive definite, t_{\min} really yields a unique minimum as seen by (3.9). Also the functional $F(\underline{X})$ is a quadratic in t as shown by (3.7). Hence it forms a parabola when plotted against t as shown in Figure 1.

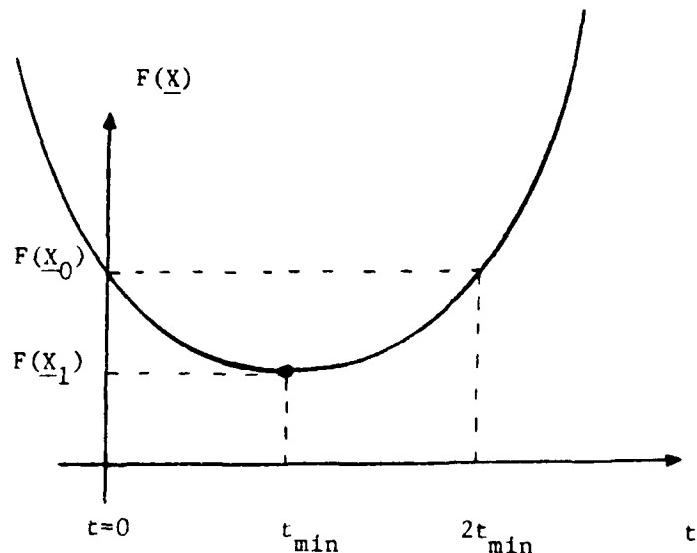


Figure 1: Plot of $F(\underline{X})$ against t .

Hence for $0 \leq t \leq 2 t_{\min}$ the value of $F(\underline{X})$ is smaller than $F(\underline{X}_0)$. The point \underline{X}_1 which is reached by moving in the direction \underline{P}_0 with $t = t_{\min}$ is then the minimum point. The decrease in $F(\underline{X})$ is given by

$$\Delta F = F(\underline{X}_1) - F(\underline{X}_0) = -\frac{1}{2} \frac{\langle \underline{R}_0, \underline{P}_0 \rangle^2}{\langle \underline{A}\underline{P}_0, \underline{P}_0 \rangle} < 0 \quad (3.10)$$

$$\text{for } \langle \underline{R}_0, \underline{P}_0 \rangle \neq 0 \quad (3.11)$$

Thus to obtain any reduction in $F(\underline{X})$ the search direction \underline{P}_0 should not be orthogonal to the residual vector \underline{R}_0 . Otherwise we remain at the trial point \underline{X}_0 .

It is also interesting to note that at the minimum point \underline{x}_1 with $t=t_{\min}$ the new residual vector \underline{R}_1 is orthogonal to \underline{P}_0 . This is because

$$\begin{aligned}\langle \underline{R}_1, \underline{P}_0 \rangle &= \langle \underline{A}\underline{x}_1 - \underline{Y}, \underline{P}_0 \rangle = \langle (\underline{A}\underline{x}_0 + t_{\min} \underline{A}\underline{P}_0 - \underline{Y}), \underline{P}_0 \rangle \\ &= \langle \underline{R}_0, \underline{P}_0 \rangle + t_{\min} \langle \underline{A}\underline{P}_0, \underline{P}_0 \rangle = 0\end{aligned}\quad (3.12)$$

For example, in the coordinate system of the unknowns, $\underline{X} = (x_1, x_2)$, the contours of $F(\underline{X}) = \text{constant}$ form concentric ellipses whose common center coincides with the minimum point of $F(\underline{X})$ and constitutes the solution point. At \underline{x}_0 , the residual \underline{R}_0 is orthogonal to the contour through the point \underline{x}_0 as it is the gradient of $F(\underline{X})$. In one iterative step we pass from \underline{x}_0 in direction \underline{P}_0 to \underline{x}_1 , where $F(\underline{X})$ is a minimum along the direction \underline{P}_0 . Here \underline{R}_1 is perpendicular to \underline{P}_0 . This is illustrated in Figure 2.

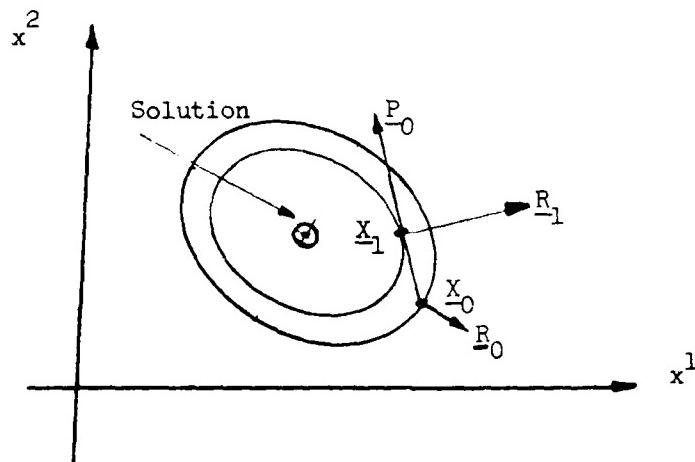


Figure 2: Interpretation of Iterative Scheme

The various iterative schemes discussed next are variations of the general iterative scheme, as the various methods differ only in the choice of the iterative direction P_i for the individual iterative steps and in the path followed (through the choice of t). An iterative method is called a stationary iterative method if the function Q_n defined as

$$\underline{x}_{n+1} = Q_n [\underline{A}, \underline{Y}, \underline{x}_n, \underline{x}_{n-1}, \dots, \underline{x}_0]$$

is independent of n . Thus in a stationary iterative method $Q_1 = Q_2 = Q_3$ and so on. Otherwise the process is called a nonstationary iterative method.

The iteration method is linear if Q_n is a linear function of $\underline{x}_n, \underline{x}_{n-1}, \dots, \underline{x}_0$.

Otherwise the method is nonlinear. In these discussions we will confine our attention to only stationary iterative methods because for nonstationary iterative methods the parameters vary with the problem.

4. LINEAR ITERATIVE METHODS

Summary

This section describes the various linear iterative schemes. It is shown that for Gauss's hand relaxation method the search directions \underline{P} are the coordinate vectors whose only non-zero element is 1 at the k th row, corresponding to the largest residual. In Jacobi's method, however, the search directions \underline{P} are chosen cyclically, i.e., starting with a vector whose only non-zero element is 1 at the first row and then gradually going down to the n th row and back to the first row again. In Seidel's process, we modify Jacobi's method by substituting a refined estimate for element x_1^i when we compute x_2^i and so on. However, the disadvantage with Seidel's process is that convergence is irregular if the largest eigenvalue of the iteration matrix is complex. This problem is remedied in the back-and-forth Seidel process. Basically it consists of two normal Seidel's processes. We proceed in the n th iteration as a normal Seidel process by computing $x_n^1 \dots x_n^N$. Then at $n + 1$ iteration we compute x_{n+1}^N to x_{n+1}^1 . The rate of convergence of linear iterative schemes can be increased by the successive overrelaxation method presented in the last subsection of this section.

4.1 GAUSS'S HAND RELAXATION METHOD [1,3,9,12,13]

This method was developed by Gauss for hand calculation and is mostly of historical significance. In this case, the elements P_n^k (from 3.6) of the search direction vector \underline{P}_n are chosen corresponding to the greatest residual in absolute value. Thus \underline{P}_n is a $N \times 1$ column vector whose only non-zero element is 1 at the k th row, where the k th row has the largest residual R_n^k in absolute value at the end of n iterations. By using (3.6) and (3.8), we arrive at the following equation in terms of the components

$$x_{n+1}^k = x_n^k - \frac{R_n^k}{A_{kk}} \cdot P_n^k \quad (4.1)$$

Thus t_{\min} for this problem is

$$t_{\min} = - \frac{\langle R_n, P_n \rangle}{\langle AP_n, P_n \rangle} = - \frac{R_n^k}{A_{kk}} \quad (4.2)$$

This method is not very suitable for automatic computation as it is a very laborious process to search for the largest absolute element in the residual.

4.2 JACOBI'S CYCLICAL ITERATION METHOD (SUCCESSIVE DISPLACEMENT METHOD) [1,3,9,12,13]

In contrast to Gauss's method, the relaxation direction \underline{P} now runs cyclically through the coordinate directions in the sequence E'_1, E'_2, \dots, E'_n regardless of the residuals. Here E'_k is a column vector with 1 at position k and zero elsewhere. This method is then equivalent to solving each of the original equations in turn for a single unknown, and hence the solution vector is changed one component at a time. Here t_{\min} is defined as

$$t_{\min} = - \frac{\langle R_n, P_n \rangle}{\langle AP_n, P_n \rangle} = - \frac{\langle R_n, E'_k \rangle}{\langle AE'_k, E'_k \rangle} = - \frac{R_n^k}{A_{kk}} \quad (4.3)$$

Hence

$$x_{n+1}^k = x_n^k - \frac{R_n^k}{A_{kk}} \cdot e_k' = \frac{1}{A_{kk}} [Y^k - \sum_{\substack{j=1 \\ j \neq k}}^N A_{kj} x_n^j]$$

Suppose A is decomposed into a diagonal matrix D (with elements $D^{ij} = A^{ii} \delta^{ij}$) a lower left triangular matrix L (with elements $L^{ij} = A^{ij}$ for $i > j$ and zero for $i \leq j$) and an upper right triangular matrix U (with elements $U^{ij} = A^{ij}$ for $i < j$ and zero for $i \geq j$). Hence

$$\underline{A} = \underline{D} + \underline{L} + \underline{U} \quad (4.4)$$

Thus the Jacobi iteration for solving $AX = Y$ takes the form

$$\underline{D}\underline{x}_{n+1} + [\underline{L} + \underline{U}] \underline{x}_n = \underline{Y}$$

or,

$$\begin{aligned} \underline{x}_{n+1} &= -[\underline{D}]^{-1} [\underline{L} + \underline{U}] \underline{x}_n + \underline{D}^{-1} \underline{Y} \\ &\triangleq \underline{G}' \underline{x}_n + \underline{H} \end{aligned} \quad (4.5)$$

The Jacobi iteration converges as long as the largest absolute eigenvalue of G' is less than unity. Other equivalent convergence conditions are described in section 8.1.

Observe that when the matrix equation $\underline{AX} = \underline{Y}$ is scaled such that $\underline{D} = \underline{I}$ (identity matrix) then

$$\underline{x}_{n+1} = \underline{x}_n - [\underline{D}]^{-1} [\underline{AX}_n - \underline{Y}] = \underline{x}_n - [\underline{D}]^{-1} \underline{R}_n = \underline{x}_n - \underline{R}_n \quad (4.6)$$

i.e. each individual component x_n^k is altered such that the residual of the k th equation is zero, without regard to the connection of the other components.

4.3 SEIDEL'S METHOD (SUCCESSIVE DISPLACEMENT METHOD) (often incorrectly called the Gauss-Seidel method) [1,3,9,12,13,14]

The rate of convergence of Jacobi's method was improved by Seidel in modifying (4.5) in the following way

$$x_{n+1}^k = \frac{1}{A_{kk}} \left[Y^k - \sum_{q=1}^{k-1} A^{kq} x_{n+1}^q - \sum_{q=k+1}^N A^{kq} x_n^q \right]$$

$$\text{or } (\underline{D} + \underline{L}) \underline{x}_{n+1} + \underline{U}\underline{x}_n = \underline{Y} \quad (4.7)$$

This is achieved by updating the column vector \underline{y}_k at each iteration so that components has been calculated and using that update in the next iteration rather than waiting for the next iteration to calculate it. Thus

$$\begin{aligned} \underline{x}_{n+1} &= -(\underline{D} + \underline{L})^{-1} \underline{U}\underline{x}_n + (\underline{D} + \underline{L})^{-1} \underline{y} \\ &\equiv \underline{x}_n + \underline{s} \end{aligned} \quad (4.8)$$

The necessary and sufficient condition for the convergence of (4.8) is that the largest eigenvalue of \underline{A} be less than unity in magnitude. Other convergence criteria will be discussed later on in section 5.1.

The major drawback of the linear iterative schemes in general and particularly that of Seidel's method is that the convergence is quite irregular if the dominant eigenvalue of \underline{Q} in (4.8) is complex. As an example consider the following problem (1.2).

Example 1: Let \underline{A} be the matrix

$$\begin{bmatrix} 1.0 & 0.7 & 0.7 & 0.2 \\ 0.7 & 1.0 & 0.7 & 0.1 \\ 0.7 & 0.7 & 1.0 & 0.1 \\ 0.2 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

and we wish to solve the matrix equation

$$\underline{A}\underline{x} = \underline{0}$$

Since $\det[\underline{A}] \neq 0$, the only possible solution is $\underline{x} = \underline{0}$. The various iterates are given in the following Table 1.

	\underline{x}_0	\underline{x}_1	\underline{x}_2	\underline{x}_3	\underline{x}_4	\underline{x}_5	\underline{x}_6	\underline{x}_7
1	1.00	-1.60	-0.83	-0.41	-0.18	-0.062	-0.0080	.01220
2	1.00	0.32	0.00	-0.12	-0.13	-0.107	-0.0761	.04927
3	1.00	0.80	0.56	0.36	0.21	0.115	.0577	.02561
4	1.00	0.21	0.11	0.06	0.03	0.012	.0034	-.0007
			\underline{x}_8		\underline{x}_9		\underline{x}_{10}	
			.01656		.014576		.010786	
			-.02953		-.016425		-008403	
			.00907		.001421		-.001527	
			-.00127		-.001514		-.001164	

Table 1: The various components of \underline{x} at the end of each iteration

i	x_1^i/x_0^i	x_2^i/x_1^i	x_3^i/x_2^i	x_4^i/x_3^i	x_5^i/x_4^i	x_6^i/x_5^i	x_7^i/x_6^i
1	-1.60	0.52	0.49	0.44	0.344	0.1290	-1.525
2	0.32	0.00	$-\infty$	1.08	0.823	0.7112	0.6474
3	0.80	0.70	0.64	0.58	0.548	0.5017	0.4438
4	0.21	0.52	0.55	0.05	0.400	0.2833	-0.0205
		x_8^i/x_7^i	x_9^i/x_8^i	x_{10}^i/x_9^i			
		1.3574	0.8802	0.739983			
		.5994	0.5562	0.511597			
		.3542	0.1567	-1.074603			
		18.1429	1.1114	0.822614			

Table 2: Ratios of x_{n+1}^i/x_n^i for Seidel's process.

In this present example it is interesting to note that the ratios do not approach any limit. This is shown in Table 2.

Such erratic behavior is to be expected since the eigenvalues of the matrix \underline{Q} in (4.8) are complex. The eigenvalues of \underline{Q}

$$-\underline{U}^{-1} [\underline{D} + \underline{L}]^{-1} [\underline{U}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & .49 & .147 & .0763 \\ 0 & .147 & .5341 & .0439 \\ 0 & .0763 & .0439 & .0228 \end{bmatrix}$$

are $\lambda = 0, 0.028333, 0.566733 \pm j.157158$. That is, the dominant eigenvalues of \underline{Q} are complex conjugates. Thus, although the Seidel scheme is convergent in this case, the convergence is erratic. This erratic behaviour of the Seidel process is remedied by the "back-and-forth" Seidel process.

4.4 BACK AND FORTH SEIDEL PROCESS [1,13]

The back-and-forth Seidel process was designed by Aitken and Rosser to overcome the irregular convergence of the Seidel process. This is achieved by making all the eigenvalues of the iterative matrix real. It proceeds as follows: Start with a first approximation vector \underline{x}_0 and then obtain $\bar{\underline{x}}_1$ by the regular Seidel process as

$$\bar{\underline{x}}_1 = -[\underline{D} + \underline{L}]^{-1} \underline{U} \underline{x}_0 + [\underline{D} + \underline{L}]^{-1} \underline{Y} \quad (4.9)$$

Then find the next iterate by applying the Siedel process to the equations in reverse order, i.e.

$$\begin{aligned} \underline{x}_1 &= -[\underline{D} + \underline{U}]^{-1} \underline{L} \bar{\underline{x}}_1 + [\underline{D} + \underline{U}]^{-1} \underline{Y} \\ &= [\underline{D} + \underline{U}]^{-1} \underline{L} [\underline{D} + \underline{L}]^{-1} \underline{U} \underline{x}_0 - [\underline{D} + \underline{U}]^{-1} \underline{L} [\underline{D} + \underline{L}]^{-1} \underline{Y} + [\underline{D} + \underline{U}]^{-1} \underline{Y} \end{aligned} \quad (4.10)$$

Thus we see that for this process the iteration matrix is

$$\underline{S} = [\underline{D} + \underline{U}]^{-1} \underline{L} [\underline{D} + \underline{L}]^{-1} \underline{U} \quad (4.11)$$

Since \underline{A} is assumed to be symmetric $\underline{L} = \underline{U}^T$ and $\underline{U} = \underline{L}^T$ (here T denotes transpose) and so the iteration matrix can be rewritten as

$$\begin{aligned}\underline{S} &= [\underline{D} + \underline{U}]^{-1} \underline{U}^T [\underline{D} + \underline{U}^T]^{-1} \underline{U} \\ &= [\underline{D} + \underline{U}]^{-1} \underline{U}^T [(\underline{D} + \underline{U})^{-1}]^T \underline{U} = \underline{M} \underline{M}^T\end{aligned}\quad (4.12)$$

where $\underline{M} = [\underline{D} + \underline{U}]^{-1} \underline{U}^T$. Thus \underline{S} is similar to a non-negative matrix.

Since $\det[\underline{M}] = \det[\underline{U}] = 0$, $\underline{M} \underline{M}^T$ is semi-definite. So we can conclude that all the eigenvalues of \underline{S} are on the real half line $x \geq 0$ and hence the dominant eigenvalue of \underline{S} is unique, though possibly may be a multiple root.

Example 2: We now apply the back-and-forth Seidel process to example

1. We again start with the same initial guess. The results are summarized in table 3

i	\underline{x}_0	\underline{x}_1	\underline{x}_2	\underline{x}_3	\underline{x}_4	\underline{x}_5	\underline{x}_6	
1	1.0	-1.60	-0.99	-0.99	-0.64	-0.64	-0.42	-0.417
2	1.0	0.32	0.48	0.06	0.23	-0.01	0.12	-0.032
3	1.0	0.80	0.88	0.63	0.64	0.43	0.45	0.305
4	1.0	0.21	0.21	0.13	0.13	0.09	0.09	0.056
	\underline{x}_4	\underline{x}_5	\underline{x}_5	\underline{x}_6	\underline{x}_6			
	-0.277	-0.27650	-0.18437	-0.18437	-0.12332			
	0.070	-0.02835	0.04305	-0.02144	0.02746			
	0.309	0.20780	0.20966	0.14033	0.14157			
	0.056	0.03736	0.03736	0.02499	0.2499			

Table 3: Various iterates of "back-and-forth" Seidel process

The ratios $\underline{x}_{n+1}^i / \underline{x}_n^i$ are obtained as

i	x_1^i/x_0^i	x_2^i/x_1^i	x_3^i/x_2^i	x_4^i/x_3^i	x_5^i/x_4^i	x_6^i/x_5^i
1	-0.99	0.65	0.66	0.660	0.66560	0.6687
2	0.48	0.48	0.52	0.583	0.61500	0.63787
3	0.88	0.73	0.70	0.687	0.67851	0.67524
4	0.21	0.62	0.70	0.622	0.66714	0.66890

Table 4: Ratios of x_{n+1}^i/x_n^i for "back-and-forth" Seidel process

The results from table 4 indicate that these ratios are tending to 0.67. i.e. that the dominant eigenvalue of \underline{S} is about 0.67. A simple calculation reveals that the eigenvalues of \underline{S} in this case are 0.65611, 0.36368, 0.02720 and 0. The dominant eigenvalue is real and is equal to the ratio of

$$\lim_{n \rightarrow \infty} \left\{ \frac{x_{n+1}^i}{x_n^i} \right\}$$

In this example, the convergence of the "back-and-forth" Seidel process, while slower than that of the ordinary Seidel process, is much more regular than the ordinary Seidel process. However, the "back-and-forth" Seidel process can easily be accelerated. It is not certain, however, which process would give the most accuracy per unit of labor.

Also, in most method of moments problems, we encounter a matrix \underline{A} whose eigenvalues are often complex. The use of the "back-and-forth" Seidel process in these problems will be justified; but, as we shall show later, there are faster schemes to treat these problems.

4.5 THE METHOD OF SUCCESSIVE OVER/UNDER RELAXATION {1.9,12,13,14,15,16}

In the case of large systems of equations, the Jacobi or Seidel process converges poorly when the maximum absolute eigenvalue (often referred to as the spectral radius) of the iteration matrix \underline{G}' in (4.5) or \underline{Q} in (4.3) lies close to unity. Convergence of the Seidel process could be improved if

instead of just reaching the minimum point at t_{\min} of Figure 1 we go beyond this point by a certain amount. It seems paradoxical at first to refrain from minimizing the quadratic functional at each iteration step with the goal of achieving better convergence. Instead of choosing $t = t_{\min}$, we choose $t = \omega t_{\min}$, where ω is a factor which may or may not change with each iteration. Thus (4.7) is now modified in the following way:

$$(\underline{D} + \underline{L}) \underline{x}_{n+1} + \underline{Ux}_n = \underline{Y}$$

$$\text{or } \underline{D} (\underline{x}_{n+1} - \underline{x}_n) = \underline{Y} - \underline{Ux}_n - \underline{DX}_n - \underline{LX}_{n+1}$$

We now introduce the parameter ω and define the new iteration

$$\omega^{-1} \underline{D} (\underline{x}_{n+1} - \underline{x}_n) = \underline{Y} - \underline{Ux}_n - \underline{DX}_n - \underline{LX}_{n+1}$$

$$\text{or } \underline{x}_{n+1} = [\underline{I} - (\omega^{-1} \underline{D} + \underline{L})^{-1} \underline{A}] \underline{x}_n + (\omega^{-1} \underline{D} + \underline{L})^{-1} \underline{Y}$$

In this case the iteration matrix is $[\underline{I} - (\omega^{-1} \underline{D} + \underline{L})^{-1} \underline{A}]$ where \underline{I} is the identity matrix. We are now interested in determining ω so as to give this matrix a small maximum eigenvalue. It is interesting to note that in symmetric definite systems of equations, the relaxation methods converge to the solution for any fixed value of ω in the range $0 < \omega < 2$. This probably could be expected intuitively as the quadratic functional is reduced in value for $0 < t < 2t_{\min}$. For $0 < \omega < 1$, the method is referred to as underrelaxation and for $1 < \omega < 2$, the method is called overrelaxation. It has been observed by Kahan and Young [14-16] that values of $\omega < 1$ tend to reduce the rate of convergence whereas $\omega > 1$ accelerates the rate of convergence. For $\omega < 1$ we overcorrect the solution vectors and hence we speak of overrelaxation methods. Unfortunately, for a given problem it is difficult to find the optimum choice of the relaxation parameter ω . For this, additional information

about the structure of matrix A is necessary. Nonetheless, we can say that the worse the condition of the matrix, the closer the optimum value of ω lies to 2. In such a case, at each iterative step we jump far beyond the minimum point to a new approximation which leaves the quadratic functional $F(\underline{X})$ almost as large as it was before. Hence the strategy of making the best improvement in each individual iterative step by going to the minimum point is not the best way of achieving the optimum long-term result.

The successive overrelaxation method has found wide application in the solution of boundary value problems by the finite difference method. In this particular type of problem one often encounters a very sparse matrix. For such special type of matrix equations optimum values of ω have been given by Kahan and Craig [14-16]. In the case of a full matrix it is difficult to find the optimum ω theoretically unless there is a certain structure to the matrix. Otherwise for each individual problem the optimum value of ω has to be obtained experimentally.

5. MONTE CARLO METHOD* [1, 17-21]

Summary

In this section we apply the law of large numbers to solve a system of linear equations. A Monte Carlo method is capable of giving a rough estimate (5-10% accuracy) of the solution in a reasonable amount of time, or when the problem is too big or complex for any other method to handle.

The Monte Carlo method is applicable if $\max_i \lambda_i(T) < 1$, where $T = A - I$.

However, this can be achieved by prescaling the matrix. The method presented in this section starts with an initial guess \underline{X}_0 of the solution \underline{X} and computes various components of \underline{X} by

$$x^j = \underline{x}_0^j - \sum_{i=1}^N [\underline{A}^{-1}]^{ji} [\underline{AX}_0 - \underline{Y}]^i$$

where $[\underline{A}^{-1}]^{ji}$ represents the element corresponding to the i th column and j th row of the inverse matrix of A . This requires slightly more work than the computation of the unknown \underline{X} by

$$x^j = \sum_{i=1}^N [\underline{A}^{-1}]^{ji} [\underline{Y}]^i$$

However, the results given by the former converge much faster if the initial estimate \underline{X}_0 is reasonable.

* It has become quite widespread nowadays in mathematical literature to speak of Monte Carlo methods (plural). This is because the same problem can be solved by simulating random variables in various ways. But here we will use Monte Carlo method (singular).

5.1 SOLUTION OF A SYSTEM OF LINEAR EQUATIONS

The Monte Carlo method is a numerical method of solving mathematical problems by means of random sampling. The method was first proposed by John von Neumann and Stanislav Ulam. Even though the theoretical justification of this method has been known for a very long time, this method could not be used on any significant scale because of the manual simulation of random variables, which is often a very laborious procedure. With the advent of the electronic computer, this method has become an extremely versatile numerical technique. The Monte Carlo method is useful in any of the following situations:

- a) A quick rough estimate of the solution is desired, which is then refined by some other means. This is because the first few steps of a Monte Carlo method tend to improve results significantly, whereas many additional steps are needed to achieve a high degree of accuracy. This method is especially efficient in solving problems which require ~ 5 - 10 per cent accuracy.
- b) The problem is too big or too complex for any other methods.
- c) Just one component of the solution vector of a large system of matrix equations or only one element of the inverse of a matrix is desired. Under such circumstances it would be very impractical to solve the complete problem.

It was shown in Chapter 4 that the solution of $\underline{A}\underline{X} = \underline{Y}$ is equivalent to the iterative scheme

$$\underline{x}_{n+1} = \underline{T}\underline{x}_n + \underline{w} \quad (5.1)$$

where $\underline{w} = [\underline{I} - \underline{T}] [\underline{A}]^{-1} \underline{y}$ (5.2)

For Jacobi's method, \underline{T} and \underline{w} are defined by

$$\begin{aligned} \underline{T} &= \underline{G}' = - [\underline{D}]^{-1} [\underline{L} + \underline{U}] && \{\text{from (4.5)}\} \\ &= - [\underline{D}]^{-1} [\underline{A} - \underline{D}] && (\text{since } \underline{A} = \underline{L} + \underline{U} + \underline{D}) \\ &= \underline{I} - [\underline{D}]^{-1} \underline{A} \end{aligned} \quad (5.3)$$

and $\underline{w} = [\underline{D}]^{-1} \underline{y}$ (5.4)

For Seidel's method

$$\begin{aligned} \underline{T} &= \underline{Q} = - [\underline{D} + \underline{L}]^{-1} [\underline{U}] && \{\text{from (4.8)}\} \\ &= - [\underline{D} + \underline{L}]^{-1} [\underline{A} - \underline{D} - \underline{L}] \\ &= \underline{I} - [\underline{D} + \underline{L}]^{-1} \underline{A} \end{aligned} \quad (5.5)$$

and $\underline{w} = [\underline{D} + \underline{L}]^{-1} \underline{y}$ (5.6)

The residual corresponding to \underline{x}_n in (5.1) is denoted by \underline{E}_n and is defined as

$$\begin{aligned} \underline{E}_n &\triangleq [\underline{I} - \underline{T}] \underline{x}_n - \underline{w} \\ &= [\underline{I} - \underline{T}] [\underline{A}]^{-1} [\underline{A}\underline{x}_n - \underline{y}] = [\underline{I} - \underline{T}] [\underline{A}]^{-1} \underline{R}_n \\ &= \underline{x}_n - \underline{x}_{n+1} \end{aligned} \quad (5.7)$$

Observe that if $\underline{A}\underline{x} = \underline{y}$ is put in the form of (5.1) then $\underline{A} = \underline{I} - \underline{T}$ with proper scaling. Under this circumstance $\underline{w} = \underline{y}$ and $\underline{E}_n = \underline{R}_n = \underline{x}_n - \underline{x}_{n+1}$, i.e., the residual is equal to the negative of the improvement in the approximate solution \underline{x}_n . We also observe that

$$\underline{E}_{n+1} = \underline{T} \underline{E}_n = \{\underline{T}\}^{n+1} \cdot \underline{E}_0 \quad (5.8)$$

$$\begin{aligned}\text{Hence, } \underline{x}_{n+1} &= \underline{x}_0 - \underline{E}_0 - \underline{E}_1 - \dots - \underline{E}_n \\ &= \underline{x}_0 - [\underline{I} + \underline{T} + \{\underline{T}\}^2 + \dots + \{\underline{T}\}^n] \underline{E}_0\end{aligned}\quad (5.9)$$

This scheme converges if the magnitude of the dominant eigenvalue of \underline{T} (or the spectral radius of \underline{T} , or the matrix norm of \underline{T} denoted by $\|\underline{T}\|$) is less than unity. Under these circumstances

$$\underline{x}_{\text{exact}} = \underline{x}_0 - [\underline{I} - \underline{T}]^{-1} \underline{E}_0 \quad (5.10)$$

In order to obtain a statistical estimation of the i th component of $\underline{x}_{\text{exact}}$ denoted by $\{\underline{x}_{\text{exact}}\}^i$ we need to have an estimation of one row of $[\underline{I} - \underline{T}]^{-1}$.

The Monte Carlo Method of computing one component of $\underline{x}_{\text{exact}}$ is to play the solitaire games $\{G^{ia}\}$, $a = 1, 2, \dots, N$ simultaneously. It will be shown that each game G^{ij} has an expected payment of $\{[\underline{I} - \underline{T}]^{-1}\}^{ij} \underline{E}_0^j$. This is equivalent to one component of the matrix product in (5.10). Based on the theory of large numbers Kolmogoroff has shown that if one plays game G^{ij} repeatedly, the average payment of M successive plays will converge to $\{[\underline{I} - \underline{T}]^{-1}\}^{ij} \underline{E}_0^j$ as $M \rightarrow \infty$, for almost all sequence of plays. The rules of the games will now be expressed in terms of balls in urns, whereas a computer will use a random number generating function.

For $1 \leq i, j \leq N$ we pick probabilities $p^{ij} \geq 0$ and the corresponding "weight" factors v^{ij} subject to the conditions that

$$1) \quad \sum_{j=1}^N p^{ij} < 1 \quad \text{for } i=1, \dots, N \text{ and} \quad (5.11)$$

$$2) \quad p^{ij} v^{ij} = T^{ij} \quad (5.12)$$

where T^{ij} is the element belonging to the i th row and j th column of \underline{T} . One way of doing this is to choose $p^{ij} = |T^{ij}|$ and $v^{ij} = \text{sgn}[T^{ij}]$. By proper scaling of the matrix equations we also make sure that $\sum_{j=1}^N p^{ij} < 1$.

Now consider N urns. In each urn U^i , we put an assortment of $N + 1$ types of balls. Each ball of the j th type is marked j and will be drawn from U^i with probability p^{ij} . Thus the balls are loaded. The $(N + 1)$ th type ball is marked "STOP" and will be drawn from U^i with the stop probability p^i defined

$$p^i = 1 - \sum_{j=1}^N p^{ij} \quad (5.13)$$

The game G^{ij} is now played as follows. Draw a ball from U^i (all drawings are with replacements). If it is a stop ball, the payment is

$$G^{ii} \triangleq \frac{E^i}{p^i} \quad (5.14)$$

Otherwise the ball would carry a mark i_1 ($1 \leq i_1 \leq N$). This would entitle us to a partial payment of v^{ii_1} . We then go to urn U^{i_1} and draw a ball. This in turn tells whether one has to stop or draw again. So we follow the treasure hunt from urn to urn until a STOP ball is drawn. Say the STOP ball is drawn from urn U^j on the k th drawing. If we have arrived at U^j via the route ρ defined by $i \rightarrow i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_{k-1} \rightarrow j$, then the payment total payment is

$$\{G^{ij}\}_{\rho} = v^{ii_1} v^{i_1 i_2} \dots v^{i_{k-1} j} \cdot \frac{E^j}{p^j} \quad (5.15)$$

Thus the probability of obtaining a STOP ball via the route ρ is

$$\{p_r\}_{\rho} = p^{ii_1} p^{i_1 i_2} \dots p^{i_{k-1} j} p^j \quad (5.16)$$

Hence the expected payment (i.e. the average payment received extended over all routes ρ) would then be

$$\mathbb{E}[G^{ij}] = \sum_{\rho} \{p_r\}_{\rho} \{G^{ij}\}_{\rho} \quad (5.17)$$

where $\mathbb{E}[\cdot]$ is the expectation operator and the sum is taken over all routes ρ which originate at i and end at j . Since we assumed $p^{ij} v^{ij} = T^{ij}$ we have

$$\mathbb{E}[G^{ij}] = [\delta^{ij} + \sum_{k=1}^{\infty} \sum_{i_1=1}^N \sum_{i_2=1}^N \dots \sum_{i_{k-1}=1}^N T^{i_1 i_2 \dots i_{k-1}}] E_o^j$$

After rearranging the terms and using $[\underline{A}]^{-1} = [\underline{I} - \underline{T}]^{-1} = \sum_{n=0}^{\infty} \underline{T}^n = \underline{I} + \underline{T} + \underline{T}^2 \dots$ we get

$$\begin{aligned} \mathbb{E}[G^{ij}] &= [\underline{I}^{ij} + \sum_{k=1}^{\infty} ((\underline{T})^k)^{ij}] E_o^j \\ &= ([\underline{I} - \underline{T}]^{-1})^{ij} E_o^j \end{aligned} \quad (5.18)$$

where δ^{ij} is Kronecker delta function and \underline{I} is the identity matrix. Thus we have shown that the expected payment of the game is G^{ij} (or mathematically the expectation of the random variable G^{ij}) is indeed only one component of the i th element of the solution vector \underline{X} . To obtain the i th element of \underline{X} we need to play all the games $\{G^{ia}\}$, $a = 1, 2, \dots, N$, as $X^i = \sum_{a=1}^N \{G^{ia}\}$

Computationally the game is played in the following way. As an example, let the matrix \underline{A} be [20]

$$\underline{A} = \underline{I} - \underline{T} = \begin{bmatrix} .4 & -.2 & -.1 \\ -.2 & .5 & -.1 \\ -.1 & -.2 & +.6 \end{bmatrix}$$

$$\text{Let } \underline{Y} = \begin{bmatrix} .1 \\ .2 \\ .3 \end{bmatrix} \quad \text{So that } \underline{X} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{Now we select}$$

$$\text{the initial approximation } \underline{X}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ so that } \underline{E}_0 = \begin{bmatrix} -.1 \\ -.2 \\ -.3 \end{bmatrix} = \underline{A}\underline{X}_0 - \underline{Y}$$

$$\text{By the terms of the problem we find } \underline{T} = \begin{bmatrix} .6 & .2 & .1 \\ .2 & .5 & .1 \\ .1 & .2 & .4 \end{bmatrix}$$

and $[p] = [T]$ and $[v^{ij}] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$. Thus $[p] = [1 - \sum_{j=1}^N p^{ij}] = \begin{bmatrix} .1 \\ .2 \\ .3 \end{bmatrix}$.

In this case $N = 3$. To play the game we select a random variable ξ between 0 and 1. If j is the point to which the walk has proceeded, we find the smallest integer k for which the cumulative probability ($\sum_{r=1}^k p^{jr}$) of going from j to k is greater than ξ . We then calculate the partial contribution of the score of the step from j to k . This is equivalent to picking a ball marked k from the j th urn. However if $\sum_{r=1}^N p^{jr} \leq \xi$, the random walk stops. This implies that the picked ball marked k from j th urn is a STOP ball. If we are interested in finding any elements belonging to the second row of the inverse matrix, we choose $i = j = 2$ (say) and the random variable $\xi = 0.82$. We now observe that

$$p = p^{21} = .2$$

$$p = p^{21} + p^{22} = .7$$

$$p = p^{21} + p^{22} + p^{23} = .8$$

and hence $\sum_{r=1}^N p^{jr} \leq \xi$ and the walk has to stop. So the payment obtained is

$$G^{ij} = G^{22} = \frac{E^2}{p^2} = \frac{.2}{.2} = 1.0 \quad (5.18a)$$

Now let us pick $\xi = .15$ and let $i = j = 2$. In this case

$$p = p^{21} = .2 > \xi$$

So we are entitled to a partial payment of $v^{21} = 1$. Next pick another

random number, say $\xi = .95$. The walk will now proceed from j to some point k (i.e., we picked a ball marked j from the i th urn and we are now going to the j th urn to pick a ball). In this case, $j = 1$ and let us observe $\sum_{r=1}^k p^{jr}$. We see

$$p = p^{11} = .6$$

$$p = p^{11} + p^{12} = .8$$

$$p = p^{11} + p^{12} + p^{13} = .9 < \xi$$

Hence the total payment is

$$G^{21} = \frac{\frac{E}{c}}{p} \cdot v^{21} = \frac{.1}{.1} = 1 \quad (5.18b)$$

We thus find all the components $\{G^{ia}\}$, $i, a = 1, 2, \dots, N$. Then the i th component of the solution X is

$$x^i = \frac{1}{N} \sum_{a=1}^N G^{ia} \quad (5.19)$$

In a Monte Carlo calculation, the problem of round-off and truncation has very little effect on accuracy. The statistical variation of the result is a more important factor. Accordingly, a measure of accuracy of the result is how the mean square deviation or the variance of the payment G^{ij} about its expected value behaves. The variance σ_{ij}^2 of G^{ij} is given by [21].

$$\begin{aligned} \sigma_{ij}^2 &= E(G^{ij} - \{[\underline{I} - \underline{T}]^{-1}\}^{ij} E_o^j)^2 \\ &= E([G^{ij}]^2) - (\{[\underline{I} - \underline{T}]^{-1}\}^{ij} E_o^j)^2 \\ &= \sum_r \{p_r\}_o \cdot \frac{\{G^{ij}\}_o^2}{p} - (\{[\underline{I} - \underline{T}]^{-1}\}^{ij} E_o^j)^2 \end{aligned} \quad (5.20)$$

Thus it can be seen that σ_{ij}^2 is finite if the magnitude of the largest eigenvalue of the matrix $[p_{ij} \cdot \{G^{ij}\}_o^2]$ is less than unity. It is interesting

to observe that if $v^{ij} = 1$ and $T^{ij} \geq 0$ then (5.20) becomes

$$\sigma_{ij}^2 = \{[\underline{I} - \underline{T}]^{-1}\}^{ij} \cdot \frac{\{E_o\}^2}{p_j} \cdot [1 - p^j \{[\underline{I} - \underline{T}]^{-1}\}^{ij}] \quad (5.21)$$

which is the variance due to a binomial distribution. If the variance after K random walks is sufficiently small we have obtained a good approximation to the one element of the solution vector. In our Monte Carlo computations we find the smallest integer K such that the probability of the required solution is within the ± 3 standard deviations (σ). This is equivalent to stating that the obtained solution G^{ij} is within $G^{ij} \pm 3 \sigma_{ij}$ with 99.77% probability.

In this example since we have done two random walks, the estimates for G^{22} and G^{21} are

$$\begin{aligned} \tilde{G}^{22} &= \frac{G^{22}}{2} = 0.5 && \text{(from 5.18a)} \\ \tilde{G}^{21} &= \frac{G^{21}}{2} = 0.5 && \text{(from 5.18b)} \end{aligned}$$

In this case

$$G_{\text{exact}}^{22} = .597402 \quad \text{and} \quad G_{\text{exact}}^{21} = .168831$$

Even though \tilde{G}^{22} is close to the exact solution, \tilde{G}^{21} is not. The variance is

$$\sigma_{22}^2 = \sigma_{21}^2 = \frac{2 [\sum \{G^{22}\}^2] - \{G^{22}\}^2}{2^2} = 0.5$$

Thus G_{exact}^{ij} lies between $\tilde{G}^{ij} \pm \sigma_{ij}$. Obviously in this case σ_{22} is very large.

However after 2376 random walks the estimate is obtained as

$$\tilde{G}^{22} = .5988$$

which is in error by .0014, and the variance is 2.4×10^{-6} . In general, many random walks are often required to obtain tolerable accuracy. The amount of work required by various methods is discussed in section 6.

5.2 ERRORS IN MONTE CARLO METHOD

In a Monte Carlo calculation the problem of round-off and truncation errors has very little effect on accuracy. The statistical variation of the result is a more important factor. Accordingly, a measure of accuracy of the solution is the mean square deviation σ and the variance defined in (5.20). It is important to note that the first few random walks tend to improve the results markedly while many additional random walks are necessary to refine them. This is because in a Monte Carlo method the variance is directly proportional to $\frac{1}{W}$, where W is the number of random walks taken.

Hence, a Monte Carlo method is highly recommended where only 5 - 10% accuracy is desired. Also it may be used to obtain the initial guess for the various iterative schemes. Finally, when the problem is too large to handle by any other method, the Monte Carlo method may be the only way to solve the problem!

6. COMPARISON OF EFFICIENCIES BETWEEN MONTE CARLO METHOD, GAUSSIAN ELIMINATION AND LINEAR ITERATIVE SCHEMES [18,19]

Summary

In this section the total amount of computations required for the linear iterative methods is compared with those for direct methods and Gaussian elimination. The total amount of work required has been computed for all three methods for the two uses: 1) when only one component of the solution is desired and 2) when the total solution is desired. A table is presented at the end which summarizes all these results.

6.1 DERIVATION OF COMPUTATIONAL REQUIREMENTS

To achieve a theoretical rather than empirical comparison we shall restrict ourselves entirely to an a priori error analysis. By error we shall mean truncation error or statistical error or both at once. We have not considered any round-off error nor the effect of miscellaneous arithmetical mistakes. The error analysis and consequent appraisal of the amount of work required to achieve a given accuracy is of necessity carried out very differently for the Monte Carlo method than for the other two methods. For the Monte Carlo method it is assumed that the problem is to find only one component of the solution vector. It is recognized freely that this restriction on the comparison is a strange one. It is made because the question of efficient Monte Carlo estimation of all components of the solution simultaneously has not yet been adequately investigated. Of course, separate statistically independent estimations can be made for each of the N components of the solution. This would multiply the measure of the work by a factor of N. Even though it is quite inefficient, we shall also use it to find the size N for which it is quite efficient to find all the components of the solution with this inefficient method.

The amount of work required for a computation will be measured only by the number of multiplications required, counting a division as one multiplication. In counting multiplications, the possibility of unit or zero factors is not taken into account.

For Gaussian elimination the total amount of work required is given as

$$K_G = \frac{N^3}{3} + N^2 - \frac{N}{3} \quad [18,19] \quad (6.1)$$

where N is the rank of the matrix.

For a linear iteration scheme, the components are obtained as

$$\underline{x}_{n+1} = \underline{T}\underline{x}_n + \underline{w}$$

and since

$$\underline{x}_{\text{exact}} = \underline{T}\underline{x}_{\text{exact}} + \underline{w}$$

we have from the two equations above

$$\begin{aligned}\underline{x}_n - \underline{x}_{\text{exact}} &= \underline{T}(\underline{x}_{n-1} - \underline{x}_{\text{exact}}) \\ &= \{\underline{T}\}^2 (\underline{x}_{n-2} - \underline{x}_{\text{exact}}) \\ &= \{\underline{T}\}^n (\underline{x}_0 - \underline{x}_{\text{exact}}) \\ &= \{\underline{T}\}^n [\underline{I} - \underline{T}]^{-1} \underline{E}_0 \quad \{\text{from (5.10)}\}\end{aligned}$$

$$\leq \frac{\{\|\underline{T}\|\}\}^n}{1 - \|\underline{T}\|} \cdot \|\underline{E}_0\| \quad (6.2)$$

Thus if we require L iterations to achieve an error of ϵ between $\underline{x}_{\text{exact}}$ and \underline{x}_n , i.e. if

$$\|\underline{x}_{\text{exact}} - \underline{x}_L\| < \epsilon \quad (6.3)$$

then we have

$$\frac{\{\|\underline{T}\|\}\}^L}{1 - \|\underline{T}\|} \cdot \|\underline{E}_0\| = \epsilon \quad (6.4)$$

$$\text{or } L = 1 + \left[\frac{\log \frac{\epsilon}{\|\underline{E}_0\|} + \log (1 - \|\underline{T}\|)}{\log \|\underline{T}\|} \right] \quad (6.5)$$

where the dotted brackets represent the truncating to the next lowest integer. The logarithms can be taken to any convenient base. Thus L iterations have to be carried out to obtain an accuracy of ϵ in \underline{x} . Each iteration counts N^2 multiplications and we have computed \underline{E}_0 . This would imply that to achieve an accuracy of ϵ in only one component of $\underline{x}_{\text{exact}}$, the total number of multiplications necessary is

$$K_{L_1} = (L - 1) N^2 + N + N^2 \quad [18,19]$$

$$= (N^2 + N + N^2) \cdot \left[\frac{\log \frac{\epsilon}{||\underline{E}_0||} + \log (1 - ||\underline{T}||)}{\log ||\underline{T}||} \right] \quad (6.6)$$

The work required for computing all the components of \underline{X}_{exact} within ϵ , will be [18,19]

$$K_{L_T} = (2N^2 + N^2) \cdot \left[\frac{\log \frac{\epsilon}{||\underline{E}_0||} + \log (1 - ||\underline{T}||)}{\log ||\underline{T}||} \right] \quad (6.7)$$

For the Monte Carlo Method we find the least value K such that the i th component

$$| \frac{x_i^i}{K} - \underline{x}_{exact}^i | < \epsilon$$

with at least 95% probability. The values of K in this case for various confidence ranges are obtained as [19].

$$K_{M_1} = N^2 + N + \frac{1}{1 - ||\underline{T}||} \left\{ 1 + \left[\frac{\xi ||\underline{E}_0||^2}{\epsilon^2 (1 - ||\underline{T}||)^2} \right] \right\} \quad (6.8)$$

where $\xi = 2.0$ for 95.45% confidence level

$= 3.317$ for 99% confidence level

$= 4.5$ for 99.7% confidence level

If we use the Monte Carlo method for finding all the components of \underline{X} within ϵ we have from [19]

$$K_{M_T} = 2N^2 + \frac{N}{1 - ||\underline{T}||} \left\{ 1 + \left[\frac{\xi ||\underline{E}_0||^2}{\epsilon^2 (1 - ||\underline{T}||)^2} \right] \right\} \quad (6.9)$$

Let

$$\alpha \triangleq \left[\frac{\log \frac{\epsilon}{||\underline{E}_0||} + \log (1 - ||\underline{T}||)}{\log ||\underline{T}||} \right] \quad (6.10)$$

$$\text{and } \beta \triangleq \frac{1}{1 - ||\underline{T}||} \left\{ 1 + \left[\frac{\xi ||\underline{E}_0||^2}{\varepsilon^2 (1 - ||\underline{T}||)^2} \right] \right\} \quad (6.11)$$

To find only one component of the solution vector within ε , we conclude from (6.1), (6.6) and (6.8) that Gaussian elimination is efficient if the size N^1 of the matrix \underline{A} lies within the range

$$1 \leq N^1 \leq \frac{3\alpha + \sqrt{9\alpha^2 + 16}}{2} \quad (6.12)$$

The linear iteration scheme is efficient for N^1 within the following range

$$\frac{3\alpha + \sqrt{9\alpha^2 + 16}}{2} < N^1 \leq \left(\frac{\beta}{\alpha} \right)^{\frac{1}{2}} \quad (6.13)$$

For $N^1 > \left(\frac{\beta}{\alpha} \right)^{\frac{1}{2}}$ the Monte Carlo method is the efficient scheme

Next we compare the total amount of work required by all the three methods to find all the components of the solution \underline{x} . Comparision of (6.1), (6.7) and (6.9) reveals that Gaussian elimination is efficient if N^T lies within the following range

$$1 \leq N^T \leq \frac{3(1+\alpha) + \sqrt{9(1+\alpha)^2 + 12}}{2} \quad (6.14)$$

The linear iteration scheme is efficient within the range

$$\frac{3(1+\alpha) + \sqrt{9(1+\alpha)^2 + 12}}{2} < N^T < \frac{\beta}{\alpha} \quad (6.15)$$

and the Monte Carlo method is efficient for

$$N^T > \frac{\beta}{\alpha} \quad (6.16)$$

Observe the precarious condition of the linear iterative scheme in (6.13) and (6.15). Depending on the values of α and β it is quite possible that the linear iterative scheme may have region where it is not efficient at all!!

Next we compute the favorable ranges of the dimensionality N for the three methods for typical values of $||\underline{T}||$ and $\frac{\varepsilon}{||\underline{E}_0||}$. This is presented in table 5.

Table 5: Favorable Lower Ranges of N for the Gauss elimination, linear iteration method and the Monte Carlo method

Norm of $\ \underline{T}\ $ and measure of accuracy required.	$\ \underline{T}\ = 0.5$	$\ \underline{T}\ = 0.9$
Gaussian elimination for solution of one component.	$\frac{\epsilon}{\ \underline{E}_0\ } \approx 0.1$ $\ \underline{E}_0\ = 0.01$	$\frac{\epsilon}{\ \underline{E}_0\ } \approx 0.001$ $\ \underline{E}_0\ = 0.1$
Linear iteration for solution of one component.	$N \leq 25$	$N \leq 34$
Monte Carlo method for solution of one component level of 95.45%	$N \geq 19$	$N \geq 143$
Gaussian elimination for solution of all components	$N \leq 18$	$26 \leq N \leq 142$
Monte Carlo method for solution of one component 99% level of 99.7%	$N \geq 25$ $N \geq 28$	$N \geq 184$ $N \geq 214$
Linear iteration for solution of all components	$20 \leq N \leq 321$	$29 \leq N \leq 20001$
Monte Carlo method for all components with confidence level of 99.45%	$N \geq 322$	$N \geq 20002$
		$N \geq 1454547$
		$136 \leq N \leq 1454546$
		$203 \leq N \leq 4546$
		$N \geq 30302$
		$N \geq 22727273$
		$N \geq 532$
		$N \geq 33172$
		$N \geq 2412365$
		$N \geq 7540$
		$N \geq 502577$
		$N \geq 37693183$
		$N \geq 721$
		$N \geq 45002$
		$N \geq 3272729$
		$N \geq 10228$
		$N \geq 681820$
		$N \geq 51136365$

Note that for $\|\underline{T}\| = 0.9$ and $\frac{\epsilon}{\|\underline{E}_0\|} = 0.1$ the linear iterative methods always require more work than the other two methods. Secondly, as the requirements of accuracy are increased the breakeven point for the Monte Carlo method also increases significantly. Thus the Monte Carlo method is quite suitable for use when we have a well-conditioned matrix and about 10% accuracy is required in the solutions. It is important to note that we have used the inefficient Monte Carlo method to compute all the components of the solution. Also we have not taken into account the effect of round-off errors in the table just presented.

If we are willing to start with the initial guess $\underline{X} = \underline{0}$, then from (6.9) the amount of work required for a specified accuracy varies as the first power of N . IF ONE IS INTERESTED ONLY IN ONE COMPONENT OF THE SOLUTION, THEN FROM (6.8) THE WORK REQUIRED BY THE MONTE CARLO METHOD BECOMES INDEPENDENT OF N TO ACHIEVE A GIVEN ACCURACY ξ .

7. NONLINEAR ITERATIVE SCHEMES [1,9,10,11]

Summary

Here we present the various nonlinear schemes as variations of the general iterative process as described in section 3. We also show how Newton's method is modified to become a steepest descent method for the solution of $\underline{AX} = \underline{Y}$. Then we discuss the conjugate direction methods of which the conjugate gradient method is treated in detail. Unlike the linear iterative methods, Monte Carlo method and the method of steepest descent, the conjugate gradient method yields the solution theoretically at the end of a finite number of steps which depend only on the distribution of the eigenvalues of \underline{A} .

7.1 HISTORY OF NONLINEAR ITERATIVE SCHEMES

Here we briefly discuss the history of nonlinear iterative schemes. In nonlinear iterative methods the refined estimate is no longer a linear function of the past estimates. Newton's method because of its quadratic convergence { $||\underline{x}_n - \underline{x}_0|| \leq c ||\underline{x}_{n-1} - \underline{x}_0||^2$ }, is mathematically the most preferred of the several known nonlinear methods for the solution of systems of equations. Practically, however, a very important limitation on Newton's method is that it does not generally converge to some solution for an arbitrary starting point. Thus Newton's method may fail to converge if the initial estimate is not sufficiently close to the solution.

The size of the domain of convergence depends upon the system of equations. For real algebraic equations, the size of the domain of convergence is generally inversely related to the degree and the number of equations. Therefore one finds that for two simultaneous second degree equations almost any initial estimate will lead to one of the solutions, while for eight simultaneous tenth degree equations the domain becomes much smaller, and it may be very difficult to obtain an initial estimate for which the iteration converges. Kantorovich thus modified Newton's method for optimization problems to become a rapidly converging descent method. Suppose again as in (3.1) we seek to minimize the functional $F(\underline{X})$ given by (3.3). This might be accomplished by the ordinary Newton method for solving the nonlinear equation $f(\underline{X}) = 0$, where $f'(\underline{X}) = F(\underline{X})$. The method is now modified to become Kantorovich's descent method. This is done by selecting the direction vectors according to Newton's method but moving along them to a point that minimizes $f(\underline{X})$ in that direction.

Thus the general iteration formula is

$$\begin{aligned}\underline{x}_{n+1} &= \underline{x}_n - \alpha_n [F'(\underline{x}_n)]^{-1} F(\underline{x}_n) \\ &= \underline{x}_n - \alpha_n [f''(\underline{x}_n)]^{-1} f'(\underline{x}_n)\end{aligned}\quad (7.1)$$

and α_n is chosen to minimize $f(\underline{x}_{n+1})$.

7.1 METHOD OF STEEPEST DESCENT [1,9,22,23,24]

In the various linear iteration schemes discussed so far, the direction vector P has been chosen as one of the coordinate axis vectors as long as P is not orthogonal to the residual R corresponding to a given trial vector X . This has been shown in (3.10). The problem is to minimize R and hence the gradient of the quadratic functional $F(X)$. Hence it is only natural to use the gradient of $F(X)$ at the approximation point to establish the relaxation direction. This is because the gradient of $F(X)$ which is R increases locally in the most rapid manner. The iteration methods using either the current or even the past residual vectors are called gradient methods.

In the method of steepest descent, the relaxation direction for the n th iteration is defined by the negative of the residual vector.

$$P_n = -R_{n-1} \quad \text{for } n = 1, 2, \dots \quad (7.2)$$

This direction is followed to the minimum point. By (3.8) the parameter t_{\min} is given by

$$t_{\min} = -\frac{\langle R_{n-1}, R_{n-1} \rangle}{\langle A R_{n-1}, R_{n-1} \rangle} \quad (7.3)$$

The value $RQ(X) = \frac{\langle AX, X \rangle}{\langle X, X \rangle}$ for $X \neq 0$ is called the Rayleigh quotient for the vector X . Observe that t_{\min} is the inverse Rayleigh quotient of the residual R_{n-1} .

In summary, the steepest descent method generates the various iterates

according to

$$\begin{aligned} \underline{x}_{n+1} &= \underline{x}_n + t_{\min} \underline{R}_n \\ &= \underline{x}_n - \frac{\langle \underline{R}_n, \underline{R}_n \rangle}{\langle \underline{A}\underline{R}_n, \underline{R}_n \rangle} \underline{R}_n \end{aligned} \quad (7.4)$$

where $\underline{R}_n = \underline{A}\underline{x}_n - \underline{y}$

From a geometric point of view, the steepest descent method involves describing a piecewise linear path with right angled corners in an N-dimensional Euclidean space, with the path terminating at the minimum of the quadratic functional $F(\underline{X})$. This is illustrated in figure 3. Unfortunately, it turns out that despite the choice of the best local direction along the largest reduction of $F(\underline{X})$ in each iteration, convergence is not good in general. This is illustrated by solving the same problem as presented in example 2 of section 4.

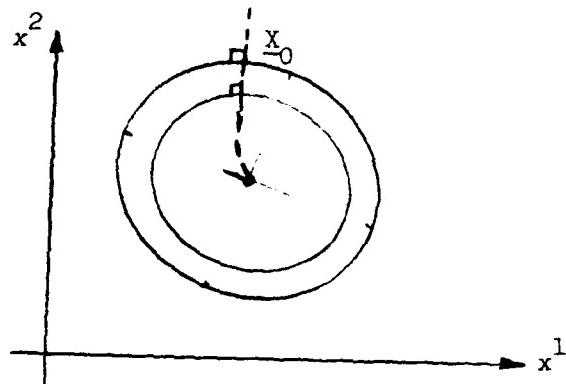


Figure 3: Principle of Steepest Descent

The various iterates are shown in Table 6.

i	\underline{x}_0	\underline{x}_1	\underline{x}_2	\underline{x}_3	\underline{x}_4	\underline{x}_5	\underline{x}_6
1	1.0	-.08125	-.02817	-.04386	-.01563	-.01713	-.00997
2	1.0	-.03967	.04264	.01892	.01358	.00739	.00690
3	1.0	-.03967	.04264	.01892	.01358	.00739	.00690
4	1.0	.41779	.02524	.01373	-.00251	-.00997	.00202
		\underline{x}_7	\underline{x}_8	\underline{x}_9	\underline{x}_{10}		
		-.01002	-.00584	-.00587	-.00342		
		.00408	.00404	.00238	.00237		
		.00408	.00404	.00238	.00237		
		.00122	.00119	.00072	.00069		

Table 6: Results of various iterations by method of steepest descent

Note that the method of steepest descent converges much faster than either of the Seidel methods. However, the tactic of seeking the most efficient goal by choosing the best local option does not lead to the best overall strategy.

The rates of convergence of the method of steepest descent is discussed in section 8.

7.2 CONJUGATE DIRECTION METHOD [1,9,20,23,24,25,26]

Conjugate direction methods are based on the generation of a set of A-orthogonal vectors and then minimizing successively in the direction of each of them. A set of vectors $\{\underline{P}_n\}$, $n = 1, 2, \dots, N$ is chosen so as to be A-conjugate, or A-orthogonal if they satisfy

$$\langle A\underline{P}_i, \underline{P}_j \rangle = 0 \quad \text{for } i \neq j. \quad (7.5)$$

Geometrically the method of conjugate directions is equivalent to that of finding the center of an N-dimensional ellipsoid when the starting point is on the surface of the ellipsoid. Thus the center point of the ellipsoid

lies on a line parallel to a fixed non null vector \underline{p}_k which is on the $(N-1)$ dimensional hyperplane

$$\langle \underline{p}_k, \underline{A}\underline{X} - \underline{Y} \rangle = 0 \quad (7.6)$$

whose normal is $\underline{A}\underline{p}_k$. This $(N-1)$ dimensional plane contains the minimum point $\underline{x}_{\text{exact}} = \underline{A}^{-1}\underline{Y}$ of the ellipsoid in the given space and is said to be conjugate to the vector \underline{p}_k . Thus the conjugate direction methods are finite step methods. That is, theoretically they all yield the exact solutions at the end of a finite number of steps ($\leq N$), assuming no truncation and round-off error.

The finite number of steps are equivalent to the number of independent eigenvalues of \underline{A} provided the dependent eigenvalues do not constitute a Jordan canonical form. Thus if the eigenvalues are equal, \underline{A} is proportional to an identity matrix and hence convergence would be obtained in one step.

But the conjugate direction method does not specify how to compute the vector \underline{p}_k : when the vectors \underline{p}_k are obtained by A -orthogonalization of the unit coordinate vectors this particular conjugate direction method yields the popular Gaussian elimination. When the vectors \underline{p}_k are obtained by A -orthogonalization of the residual vectors \underline{R}_k , a conjugate gradient method results.

The conjugate gradient method applies more constraints on the iteration process than those imposed by Gaussian elimination. Hence the conjugate gradient method may yield acceptable results under conditions when Gaussian elimination fails. This has been illustrated in section 2.5.

7.3 CONJUGATE GRADIENT METHOD [1,9,23,24,25,26]

For the solution of $\underline{A}\underline{X}=\underline{Y}$, the conjugate gradient method starts with an initial guess \underline{x}_0 and obtains

$$\underline{P}_0 = -\underline{R}_0 = \underline{Y} - \underline{A}\underline{X}_0 \quad (7.7)$$

and then develops each successive approximation by

$$\underline{x}_{n+1} = \underline{x}_n + t_n \underline{z}_n \quad (7.8)$$

where $t_n = - \frac{\langle \underline{P}_n, \underline{R}_n \rangle}{\langle \underline{A}\underline{P}_n, \underline{P}_n \rangle}$ (7.9)

This value of t_n takes $\underline{F}(\underline{X}_n)$ to a minimum point in the n th iteration. Next the residuals are generated iteratively by

$$\underline{R}_{n+1} = \underline{R}_n + t_n \cdot \underline{A}\underline{P}_n \quad (7.10)$$

and the direction vectors are obtained iteratively as

$$\underline{P}_{n+1} = -\underline{R}_{n+1} + q_n \underline{P}_n \quad (7.11)$$

where q_n is defined as

$$q_n = \frac{\langle \underline{A}\underline{P}_n, \underline{R}_{n+1} \rangle}{\langle \underline{A}\underline{P}_n, \underline{P}_n \rangle} \quad (7.12)$$

Thus in order to arrive at \underline{X}_n from \underline{X}_{n-2} in the conjugate gradient method we go first to the minimum point along \underline{P}_{n-2} to \underline{X}_{n-1} and then travel along \underline{P}_{n-1} which is A-conjugate to \underline{P}_{n-2} . The directions \underline{P}_n are A-conjugate and the residuals \underline{R}_n form an orthogonal system. Hence the method of conjugate gradients yields the solution in at most M steps, where M is the number of independent eigenvalues of the matrix A , provided these eigenvalues do not constitute a Jordan canonical form.

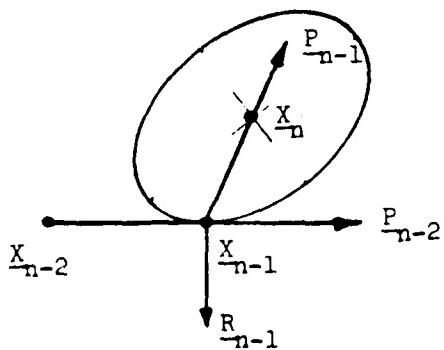


Figure 4: Method of Conjugate gradient

A more convenient form of computation may be derived for (7.9), (7.11) and (7.12). It is seen that

$$\langle \underline{R}_n, \underline{P}_n \rangle = - \langle \underline{R}_n, \underline{R}_n \rangle + q_n \langle \underline{R}_n, \underline{P}_{n-1} \rangle = - \|\underline{R}_n\|^2$$

since $\langle \underline{R}_n, \underline{P}_{n-1} \rangle = 0$ (see figure 4). Thus

$$t_n = \frac{\|\underline{R}_n\|^2}{\langle \underline{A}\underline{P}_n, \underline{P}_n \rangle} \quad (7.13)$$

Also analogous to the method of steepest descent two successive residual vectors are orthogonal as from above $\langle \underline{R}_n, \underline{P}_{n-1} \rangle = 0$ and $\underline{P}_n = -\underline{R}_{n-1}$ (from 7.2)

$$\langle \underline{R}_i, \underline{R}_j \rangle = 0 \quad \text{for } i \neq j \quad (7.14)$$

Since $\underline{P}_0, \underline{P}_1, \dots, \underline{P}_i$ are obtained by computing a set of A-orthogonal vectors from $\underline{R}_0, \underline{R}_1, \dots, \underline{R}_i$ we have

$$\begin{aligned} \langle \underline{P}_i, \underline{A}\underline{P}_j \rangle &= 0 && \text{for } i \neq j \\ \langle \underline{R}_i, \underline{A}\underline{P}_k \rangle &= 0 && \text{for } i < k \\ \langle \underline{A}\underline{P}_{k-1}, \underline{R}_i \rangle &= 0 && \text{for } i > k \\ \langle \underline{P}_i, \underline{A}\underline{P}_i \rangle &= - \langle \underline{R}_i, \underline{A}\underline{P}_i \rangle \\ \langle \underline{P}_i, \underline{R}_j \rangle &= \langle \underline{P}_i, \underline{R}_i \rangle = - \langle \underline{R}_i, \underline{R}_i \rangle && \text{for } j \leq i \\ \langle \underline{P}_i, \underline{R}_j \rangle &= 0 && \text{for } j > i \end{aligned} \quad (7.15)$$

Also from equations (7.10) to (7.14) we have

$$\begin{aligned} \langle \underline{R}_{n+1}, \underline{A}\underline{P}_n \rangle &= \langle \underline{R}_{n+1}, \frac{\underline{R}_{n+1} - \underline{R}_n}{t_n} \rangle = \frac{\|\underline{R}_{n+1}\|^2}{t_n} \\ &= \frac{\|\underline{R}_{n+1}\|^2}{\|\underline{R}_n\|^2} \cdot \langle \underline{A}\underline{P}_n, \underline{P}_n \rangle \end{aligned}$$

Hence

$$q_n = \frac{\langle \underline{AP}_n, \underline{R}_{n+1} \rangle}{\langle \underline{P}_n, \underline{AP}_n \rangle} = \frac{\|\underline{R}_{n+1}\|^2}{\|\underline{R}_n\|^2} \quad (7.16)$$

Equations (7.7), (7.8), (7.10), (7.11), (7.13) and (7.16) of the conjugate gradient method are applied to solve the same problem presented in example 1.

The various iterates for the solution are shown in table 7.

i	\underline{x}_0	\underline{x}_1	\underline{x}_2	\underline{x}_3	\underline{x}_4
1	1.0	- .08125	- .04848	.6012 $\times 10^{-5}$.7966 $\times 10^{-7}$
2	1.0	- .03966	.02373	.6217 $\times 10^{-5}$.2905 $\times 10^{-6}$
3	1.0	- .03966	.02373	.6217 $\times 10^{-5}$.2905 $\times 10^{-6}$
4	1.0	.41779	.00599	.1281 $\times 10^{-5}$	-.9905 $\times 10^{-6}$

Table 7: Results of various iterations given by the conjugate gradient method

Observe that there is a sharp increase in the accuracy of the solutions at \underline{x}_3 . One has obtained essentially an exact solution after 3 iterations. This is because the four eigenvalues of the matrix \underline{A} are 2.4372, .9725, .300 and .2903. Note that there are approximately 3 independent eigen values of \underline{A} . Thus one would expect excellent results at the end of 3 steps. Hence the conjugate gradient method might converge quite fast for a large system of equations if the matrix has quite a few eigenvalues bunched together. This generally happens in matrices which have dominant diagonals (as in the magnetic field integral equation).

Next we derive the various theoretical rates of convergence of the various iterative schemes and show how the method of conjugate gradient converges much faster than others.

8. ANALYSIS OF CONVERGENCE OF VARIOUS ITERATIVE SCHEMES

Summary

The rates of convergence of the various iterative schemes, both linear and nonlinear, are discussed in this section. We show that for the linear iterative schemes, the rate at which the \underline{x}_n 's approach the exact solution is linear and the \underline{x}_n 's converge geometrically with the ratio $|\lambda_1|$ only in an asymptotic sense, where λ_1 is the largest eigenvalue of the iteration matrix. The nonlinear iterative schemes on the other hand have a geometrical rate of convergence to begin with and possess superlinear convergence when A is a Legendre operator. For the method of steepest descent the ratio for the geometric convergence is $\frac{\text{cond } [A] - 1}{\text{cond } [A] + 1}$. For the method of conjugate gradient (which, unlike the other iterative methods, yields the solution in a finite number of steps) the minimum rate of convergence is given by the ratio $\frac{\sqrt{\text{cond } [A]} - 1}{\sqrt{\text{cond } [A]} + 1}$. The J steps steepest descent method (equivalent to taking J steps of the steepest descent simultaneously) has the same rate of convergence as the conjugate gradient method.

8.1 RATE OF CONVERGENCE FOR THE LINEAR ITERATIVE SCHEMES [9]

For the linear iterative scheme we have

$$\underline{X}_{n+1} = \underline{T} \underline{X}_n + \underline{W} \quad (8.1)$$

as given by (5.1). For (3.1) to converge, a necessary and sufficient condition is that the magnitude of the dominant eigenvalue of the iterative matrix \underline{T} be less than unity. To prove this we define the error vector \underline{ER} for the n th iterate as

$$\underline{ER}_n = \underline{X}_{\text{exact}} - \underline{X}_n \quad (8.2)$$

Since

$$\underline{X}_{\text{exact}} = \underline{T} \underline{X}_{\text{exact}} + \underline{W} \quad (8.3)$$

we have

$$\underline{ER}_{n+1} = \underline{T} \cdot \underline{ER}_n = \{\underline{T}\}^n \cdot \underline{ER}_0$$

Thus

$$||\underline{ER}_{n+1}|| = ||\{\underline{T}\}^n \cdot \underline{ER}_0|| \leq ||\{\underline{T}\}^n|| \cdot ||\underline{ER}_0|| \leq \{||\underline{T}||\}^n \cdot ||\underline{ER}_0|| \quad (8.4)$$

Under the premise $||\underline{T}|| < 1$ (i.e., the magnitude of the dominant eigenvalue is less than unity) it follows from (8.4) that

$$\lim_{n \rightarrow \infty} ||\underline{ER}_n|| = 0 \quad (8.5)$$

and thus \underline{X}_n converges to the solution $\underline{X}_{\text{exact}}$.

Next we show that the dominant eigenvalue of T dictates the rate of convergence of the linear iterative process.

Assume for the sake of simplicity that the matrix which is neither symmetric nor positive has N independent eigenvectors $\underline{v}_1, \underline{v}_2, \dots, \underline{v}_N$ with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_N$. The error vector \underline{ER}_0 can thus be represented as a

linear combination of the eigenvectors

$$\underline{\underline{ER}}_m = \sum_{i=1}^N c_i \underline{v}_i \quad (8.6)$$

For the mth error vector, we have [9]

$$\underline{\underline{ER}}_m = \sum_{i=1}^N c_i \{\lambda_i\}^m \underline{v}_i \text{ for } m = 1, 2, \dots \quad (8.7)$$

This equation permits us to make a qualitative statement about the asymptotic convergence behavior of $\underline{\underline{ER}}_m$. The dominant eigenvalue λ_1 of matrix $\underline{\underline{T}}$, that is, the one with the largest magnitude, generally governs the rate of convergence, since by (8.7) the smaller eigenvalues $\{\lambda_i\}^m$ approach zero faster than $\{\lambda_1\}^m$ with increasing m . Thus every vector norm $\|\underline{\underline{ER}}_m\|$ converges asymptotically to zero like a geometric series with the ratio $|\lambda_1|$.

So a linear iterative scheme converges linearly, and converges geometrically only in an asymptotic sense. Let a sequence $\{s_n\}$ converge to α . Then the sequence $\{s_n\}$ is said to have linear convergence if

$$e_{i+1} = (\beta + \sigma_i) e_i \quad (8.8)$$

where $e_i = \alpha - s_i$ and for a constant β , $|\beta| < 1$ and $\sigma_i \rightarrow 0$ as $i \rightarrow \infty$.

The sequence $\{s_n\}$ is said to have geometric convergence if

$$e_{i+1} = \beta e_i \quad (8.9)$$

$|\beta| < 1$. Thus geometric convergence is a special case of linear convergence in which all $\sigma = 0$. For a large number of iterations the linear iterative schemes converge geometrically since for sufficiently large m and $k > 0$

$$\|\underline{x}_{m+k} - \underline{x}_0\| < \{\lambda_1\}^k \|\underline{x}_m - \underline{x}_0\| \quad (8.10)$$

Thus the smaller the dominant eigenvalue of $\underline{\underline{T}}$, the faster the convergence.

Conversely when the magnitude of the dominant eigenvalue is close to one, many

iterations are necessary. Hence from (8.10) the number of iterations necessary to reduce the error $\|\underline{x}_m - \underline{x}_0\|$ by a factor of 10 is approximately inversely proportional to $-1/\{\log_{10} \lambda_1\}$. Thus to gain an additional significant decimal place in \underline{x}_m we need k iterations.

So the fastest rate of convergence that can be achieved by the linear iterative schemes can at best be geometric and the successive approximations always converge for a definite system of equations. Equivalently the latter condition may also be stated by saying $\|\underline{T}\| < 1$. This condition of convergence may also be stated in several different ways. In order that the linear iteration schemes converge for every \underline{x}_0 and for any order N of the equations $\underline{A}\underline{x} = \underline{y}$ it is necessary and sufficient that any of the following conditions hold (conditions 3-8 describe the diagonal dominance of the matrices):

$$1. \quad \{\underline{T}\}^k \rightarrow 0 \quad \text{as } k \rightarrow \infty$$

$$2. \quad \text{the magnitude of the dominant eigenvalue of } \underline{T}, \text{ i.e. } |\lambda_1(\underline{T})|, \text{ be less than unity.}$$

$$3. \quad \sum_{k=1}^N \sum_{i=1}^N \left[\frac{a_{ik}}{a_{ii}} \right]^2 < 1$$

$i \neq k$

[Note: this condition is not valid for Siedel's method. This is valid for Jacobi's method only.] {Theorem of E. Schmidt - Mises - Geiringer}

$$4. \quad \sum_{\substack{m=1 \\ k \neq m}}^N \left| \frac{a_{km}}{a_{kk}} \right| < 1 \quad \text{for all } k=1, \dots, N$$

{Theorem of Frobenius - Mises - Geiringer}

$$5. \sum_{\substack{k=1 \\ k \neq m}}^N \left| \frac{A_{km}}{A_{kk}} \right| < 1 \quad \text{for all } m=1, \dots, N$$

{Theorem of Frobenius - Mises - Geiringer}

$$6*. \sum_{\substack{i=1 \\ i \neq k}}^N \left| \frac{A_{ik}}{A_{ii}} \right| \leq 1 \quad \text{for all } i = 1, \dots, N$$

with $\sum_{\substack{k=1 \\ i \neq k}}^N \left| \frac{A_{ik}}{A_{ii}} \right| < 1$ for at least one value of i

and A is irreducible

$$7*. \sum_{\substack{i=1 \\ i \neq k}}^N \left| \frac{A_{ik}}{A_{ii}} \right| \leq 1 \quad \text{for all } k = 1, \dots, N$$

with $\sum_{\substack{i=1 \\ i \neq 1}}^N \left| \frac{A_{ik}}{A_{ii}} \right| < 1$ for at least one value of k

and A is irreducible

* It is important to note that for conditions 6 and 7, there is a further restriction on the matrix A. A must be an irreducible matrix, i.e., a matrix which cannot be put in the form $\begin{bmatrix} P & Q \\ 0 & R \end{bmatrix}$ (where R and P are square) by simultaneous row and column permutations.

$$8. \quad \left(\frac{t+1}{u+1}\right)^N \geq \left(\frac{1}{u}\right) \quad \text{according as } t \geq u$$

$$\text{where } t = \max_{i>j} \left| \frac{A^{ij}}{A^{ii}} \right|, \quad u = \max_{i<j} \left| \frac{A^{ij}}{A^{ii}} \right|$$

{Theorem of Stein - Rosenberg}

The conditions 3-8 have to do with the diagonal dominance of the matrices.

8.2. RATE OF CONVERGENCE FOR NONLINEAR ITERATIVE SCHEMES

8.2.1. METHOD OF STEEPEST DESCENT [22,26-30]

Let us assume \underline{A} is symmetric positive definite such that the eigenvalues λ_i of \underline{A} may be arranged as

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N > 0 \quad (8.11)$$

Let $\underline{v}_1, \underline{v}_2, \dots, \underline{v}_n$ be the respective orthonormalized eigenvectors corresponding to the eigenvalues λ_i . Then if \underline{z} is an arbitrary vector, it can be represented as

$$\underline{z} = \beta_1 \underline{v}_1 + \beta_2 \underline{v}_2 + \dots + \beta_N \underline{v}_N$$

when β_i are constants and

$$\langle \underline{A}\underline{z}, \underline{z} \rangle = \beta_1^2 \lambda_1 + \beta_2^2 \lambda_2 + \dots + \beta_N^2 \lambda_N \quad (8.12)$$

Thus

$$\begin{aligned} \lambda_N (\beta_1^2 + \beta_2^2 + \dots + \beta_N^2) &= \lambda_N \langle \underline{z}, \underline{z} \rangle \leq \langle \underline{A}\underline{z}, \underline{z} \rangle \\ &\leq \lambda_1 (\beta_1^2 + \beta_2^2 + \dots + \beta_N^2) = \lambda_1 \langle \underline{z}, \underline{z} \rangle \end{aligned} \quad (8.13)$$

Consequently one can find two constants $b > 0$ and $B > 0$ for \underline{A} such that

$$b \langle \underline{z}, \underline{z} \rangle \leq \langle \underline{A}\underline{z}, \underline{z} \rangle \leq B \langle \underline{z}, \underline{z} \rangle \quad (8.14)$$

Now we consider the difference $F(\underline{x}_1) - F(\underline{x}_0)$, where

$$F(\underline{x}) = \frac{1}{2} \langle \underline{A}\underline{x}, \underline{x} \rangle - \langle \underline{y}, \underline{x} \rangle. \text{ After some calculations using (7.4) we obtain}$$

$$F(\underline{x}_1) - F(\underline{x}_0) = \frac{1}{2} \frac{\{\langle \underline{R}_0, \underline{R}_0 \rangle\}2}{\langle \underline{R}_0, \underline{A}\underline{R}_0 \rangle}$$

and

$$F(\underline{x}_o) - F(\underline{x}_{\text{exact}}) = \frac{1}{2} \langle \underline{x}_o - \underline{x}_{\text{exact}}, \underline{A}(\underline{x}_o - \underline{x}_{\text{exact}}) \rangle$$

Thus

$$\frac{F(\underline{x}_1) - F(\underline{x}_o)}{F(\underline{x}_o) - F(\underline{x}_{\text{exact}})} = - \frac{\langle \underline{R}_o, \underline{R}_o \rangle^2}{\langle \underline{R}_o, \underline{A}\underline{R}_o \rangle \langle \underline{A}^{-1} \underline{R}_o, \underline{R}_o \rangle} \quad (8.15)$$

Next we expand \underline{R}_o as a series in orthonormalized eigenvectors of \underline{A} i.e.

$$\underline{R}_o = \gamma_1 \underline{v}_1 + \gamma_2 \underline{v}_2 + \dots + \gamma_N \underline{v}_N \quad (8.16)$$

where γ_i are constants. Then

$$\underline{A}\underline{R}_o = \gamma_1 \lambda_1 \underline{v}_1 + \gamma_2 \lambda_2 \underline{v}_2 + \dots + \gamma_N \lambda_N \underline{v}_N \quad (8.17)$$

$$\underline{A}^{-1} \underline{R}_o = \gamma_1 \underline{v}_1 \lambda_1^{-1} + \gamma_2 \underline{v}_2 \lambda_2^{-1} + \dots + \gamma_N \underline{v}_N \lambda_N^{-1} \quad (8.18)$$

where \underline{z}_i are the various normalized eigenvectors of \underline{A} . So,

$$\frac{F(\underline{x}_o) - F(\underline{x}_1)}{F(\underline{x}_o) - F(\underline{x}_{\text{exact}})} = \frac{(\gamma_1^2 + \gamma_2^2 + \dots + \gamma_N^2)^2}{(\gamma_1^2 \lambda_1 + \gamma_2^2 \lambda_2 + \dots + \gamma_N^2 \lambda_N)(\gamma_1^2 \lambda_1^{-1} + \gamma_2^2 \lambda_2^{-1} + \dots + \gamma_N^2 \lambda_N^{-1})} \quad (8.19)$$

Let

$$\frac{\gamma_i^2}{\gamma_1^2 + \gamma_2^2 + \dots + \gamma_N^2} = \theta_i \quad ; \quad \theta_i \geq 0 \quad \text{and} \quad \sum_{i=1}^N \theta_i = 1 \quad (8.20)$$

$$\frac{F(\underline{x}_o) - F(\underline{x}_1)}{F(\underline{x}_o) - F(\underline{x}_{\text{exact}})} = \frac{1}{(\theta_1 \lambda_1 + \theta_2 \lambda_2 + \dots + \theta_N \lambda_N)(\theta_1 \lambda_1^{-1} + \theta_2 \lambda_2^{-1} + \dots + \theta_N \lambda_N^{-1})} \quad (8.21)$$

We replace λ_i by the new variable λ_i^1 defined as

$$\lambda_i^1 = \sqrt{Bb} \lambda_i \quad (8.22)$$

So if $0 < b \leq \lambda_i \leq B$ ($i = 1, 2, \dots, N$)

then

$$\sqrt{\frac{b}{B}} \leq \lambda_i^1 \leq \sqrt{\frac{B}{b}} \quad (8.23)$$

and

$$\sum_{i=1}^N \theta_i \lambda_i \sum_{i=1}^N \theta_i \lambda_i^{-1} = \sum_{i=1}^N \theta_i \lambda_i^1 \sum_{i=1}^N \theta_i \{\lambda_i^1\}^{-1} \quad (8.24)$$

Since the geometric mean of a series is less than, or equal to the arithmetic mean, then

$$\sqrt{\left\{ \sum_{i=1}^N \theta_i \lambda_i^1 \right\} \left\{ \sum_{i=1}^N \theta_i \{\lambda_i^1\}^{-1} \right\}} \leq \left\{ \sum_{i=1}^N \theta_i (\lambda_i^1 + \frac{1}{\lambda_i^1}) \right\} \times 0.5$$

or,

$$\sqrt{\left\{ \sum_{i=1}^N \theta_i \lambda_i^1 \right\} \left\{ \sum_{i=1}^N \theta_i \lambda_i^{-1} \right\}} \leq \frac{1}{2} \left\{ \sum_{i=1}^N \theta_i \right\} \left\{ \sqrt{\frac{B}{b}} + \sqrt{\frac{b}{B}} \right\}$$

or,

$$\sum_{i=1}^N \theta_i \lambda_i \sum_{i=1}^N \theta_i \lambda_i^{-1} \leq \frac{1}{4} \left[\sqrt{\frac{b}{B}} + \sqrt{\frac{B}{b}} \right]^2 \quad (8.25)$$

Thus

$$\frac{F(\underline{x}_0) - F(\underline{x}_1)}{F(\underline{x}_0) - F(\underline{x}_{\text{exact}})} \geq \left[\sqrt{\frac{b}{B}} + \sqrt{\frac{B}{b}} \right]^2 = \xi \quad (8.26)$$

Here $0 < \xi \leq 1$. Hence

$$\begin{aligned} F(\underline{x}_1) - F(\underline{x}_{\text{exact}}) &\leq (1 - \xi) \{ F(\underline{x}_0) - F(\underline{x}_{\text{exact}}) \} \\ &\leq \left[\frac{B - b}{B + b} \right]^2 \cdot \{ F(\underline{x}_0) - F(\underline{x}_{\text{exact}}) \} \end{aligned}$$

Thus for any k

$$F(\underline{x}_k) - F(\underline{x}_{\text{exact}}) \leq \left[\frac{B - b}{B + b} \right]^{2k} \{ F(\underline{x}_0) - F(\underline{x}_{\text{exact}}) \}$$

So from (8.13)

$$\begin{aligned} \|\underline{x}_k - \underline{x}_{\text{exact}}\|^2 &= \langle \underline{x}_{\text{exact}} - \underline{x}_k, \underline{x}_{\text{exact}} - \underline{x}_k \rangle \\ &\leq \frac{1}{b} \langle A\underline{x}_k - \underline{y}, \underline{x}_k - \underline{x}_{\text{exact}} \rangle = \frac{2}{b} [F(\underline{x}_k) - F(\underline{x}_{\text{exact}})] \quad (8.27) \end{aligned}$$

Thus,

$$\|\underline{x}_k - \underline{x}_{\text{exact}}\|^2 \leq \frac{2[F(\underline{x}_0) - F(\underline{x}_{\text{exact}})]}{b} \cdot \left(\frac{B-b}{B+b}\right)^{2k} \quad (8.28)$$

So we have proved that the method of steepest descent converges geometrically to the exact solution. For the case when A has N distinct eigenvalues Akaike [31] has shown that (8.27) is the best possible estimate.

It has been shown by Daniel [26] and Hayes [11] that whenever A is a Legendre operator (i.e., A is a sum of a positive definite bounded self-adjoint operator plus a completely continuous operator) the method of steepest descent converges faster than that of a geometric series with ratio greater than $\left(\frac{B-b}{B+b}\right)^2$. This type of convergence is referred to as "superlinear" convergence.

Thus we have shown that the method of steepest descent converges at worst like a geometric series, and that in most cases the convergence is superlinear.

8.2.2 METHOD OF CONJUGATE GRADIENT [26,27,31,32]

The method of conjugate gradient generally requires a little more computation than the method of steepest descent. However this slight increase in the amount of computation required leads to a significant improvement in the rate of convergence over that of the method of steepest descent. In the conjugate gradient method we obtain

$$\underline{x}_{\text{exact}} = \underline{x}_0 + \beta_0 \underline{R}_0 + \beta_1 \underline{A}\underline{R}_0 + \beta_2 \underline{A}^2 \underline{R}_0 + \dots + \beta_{N-1} \underline{A}^{N-1} \underline{R}_0 \quad (8.29)$$

This results from the fact that

$$\begin{aligned}\underline{x}_1 &= \underline{x}_0 + \alpha_0^1 \underline{r}_0 \\ \underline{x}_2 &= \underline{x}_0 + \alpha_0^2 \underline{r}_0 + \alpha_1^2 \underline{r}_1 \\ &\vdots \\ \underline{x}_k &= \underline{x}_0 + \alpha_0^k \underline{r}_0 + \alpha_1^k \underline{r}_1 + \dots + \alpha_{k-1}^k \underline{r}_{k-1}\end{aligned}\tag{8.30}$$

and so on, where α_i^j are known constants determined by the method. Now we define $\underline{z}_1, \underline{z}_2, \dots, \underline{z}_N$ as the orthonormalized eigenvectors of the symmetric definite matrix \underline{A} , corresponding to the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$. So

$$\underline{x}_0 - \underline{A}^{-1} \underline{y} = \underline{x}_0 - \underline{x}_{\text{exact}} = \sum_{i=1}^N c_i \underline{z}_i\tag{8.31}$$

where c_i are constants. We find

$$\begin{aligned}\underline{x}_1 - \underline{A}^{-1} \underline{y} &= (\underline{I} + \gamma_0^1 \underline{A})(\underline{x}_0 - \underline{A}^{-1} \underline{y}) = \sum_{i=1}^N c_i (1 + \gamma_0^1 \lambda_i) \underline{z}_i \\ \underline{x}_2 - \underline{A}^{-1} \underline{y} &= (\underline{I} + \gamma_0^2 \underline{A})(\underline{I} + \gamma_1^2 \underline{A})(\underline{x}_0 - \underline{A}^{-1} \underline{y}) = \sum_{i=1}^N c_i (1 + \gamma_0^2 \lambda_i)(1 + \gamma_1^2 \lambda_i) \underline{z}_i \\ \underline{x}_3 - \underline{A}^{-1} \underline{y} &= (\underline{I} + \gamma_0^3 \underline{A})(\underline{I} + \gamma_1^3 \underline{A})(\underline{I} + \gamma_2^3 \underline{A})(\underline{x}_0 - \underline{A}^{-1} \underline{y}) \\ &= \sum_{i=1}^N c_i (1 + \gamma_0^3 \lambda_i)(1 + \gamma_1^3 \lambda_i)(1 + \gamma_2^3 \lambda_i) \underline{z}_i \\ \underline{x}_k - \underline{A}^{-1} \underline{y} &= \sum_{i=1}^N c_i \prod_{j=0}^{k-1} (1 + \gamma_j^k \lambda_i) \underline{z}_i\end{aligned}\tag{8.32}$$

where \underline{I} is the identity matrix and γ_j^i are known constants determined by the conjugate gradient method. Therefore

$$\begin{aligned}\|\underline{x}_k - \underline{x}_{\text{exact}}\| &= \|\underline{x}_k - \underline{A}^{-1} \underline{y}\| \leq \max_{1 \leq i \leq N} \left| \prod_{j=0}^{k-1} (1 + \gamma_j^k \lambda_i) \right| \cdot \|\underline{x}_0 - \underline{A}^{-1} \underline{y}\| \\ &\leq |\mathcal{O}_k(\lambda)| \cdot \|\underline{x}_0 - \underline{x}_{\text{exact}}\|\end{aligned}\tag{8.33}$$

where $o_k(\lambda)$ is a polynomial in λ defined as

$$o_k(\lambda) = \prod_{j=0}^{k-1} (1 + \gamma_j^k \lambda) \quad (8.34)$$

The problem is then to find a k th degree polynomial $o_k(\lambda)$ defined for $b \leq \lambda \leq B$ such that $\max_{b \leq \lambda \leq B} |o_k(\lambda)|$ is a minimum. Such a polynomial was given by W. Markoff (in 1892) and is defined as [1]

$$o_k(\lambda) = \frac{T_k \left(\frac{B+b-2\lambda}{B-b} \right)}{T_k \left[\frac{B+b}{B-b} \right]} \quad (8.35)$$

and $T_k(t) = \cos(k \arccos t)$ is the well known Chebyshev polynomial of degree k adjusted to the interval $-1 \leq t \leq 1$. Thus $o_k(\lambda)$ is a Chebyshev polynomial of degree k adjusted to the interval $b \leq \lambda \leq B$ and scaled so that $o_k(0) = 1$.

Thus from (8.35)

$$\max_{b \leq \lambda \leq B} |o_k(\lambda)| \leq \frac{1}{T_k \left[\frac{B+b}{B-b} \right]} \quad (8.36)$$

Note in this case $\frac{B+b}{B-b} > 1$ and so the Chebyshev polynomials are defined as

$$T_k(t) = \cosh(k \operatorname{arccosh} t) \quad \text{for } t > 1$$

Hence from the expansion of the Chebyshev polynomials

$$T_k \left[\frac{B+b}{B-b} \right] = \frac{1}{2} \left[\left\{ \frac{B+b}{B-b} + \sqrt{\left(\frac{B+b}{B-b} \right)^2 - 1} \right\}^k + \left\{ \frac{B+b}{B-b} - \sqrt{\left(\frac{B+b}{B-b} \right)^2 - 1} \right\}^k \right] \quad (8.37)$$

Hence from (8.33)

$$o_k(\lambda) = \left| \frac{x_k - x_{\text{exact}}}{x_0 - x_{\text{exact}}} \right| \leq \frac{2}{\left\{ \frac{\sqrt{B} + \sqrt{b}}{\sqrt{B} - \sqrt{b}} \right\}^k + \left\{ \frac{\sqrt{B} - \sqrt{b}}{\sqrt{B} + \sqrt{b}} \right\}^k} \quad (8.38)$$

$$\leq 2 \cdot \frac{\left\{ \frac{\sqrt{B} - \sqrt{b}}{\sqrt{B} + \sqrt{b}} \right\}^k}{\left\{ \frac{\sqrt{B} + \sqrt{b}}{\sqrt{B} - \sqrt{b}} \right\}^k} \quad (8.39)$$

Observe that (8.38) is a better estimate than (8.39). Also this estimate (8.38) cannot be improved upon. This fact is well known [31].

Also as before, if A is a Legendre operator (most operators defined on finite dimensional spaces are Legendre operators) then the method of conjugate gradient converges faster than a geometric series with ratio greater than $\{(\frac{\sqrt{B} - \sqrt{b}}{\sqrt{B} + \sqrt{b}})\}^2$ [31].

A better estimate than (8.38) can be obtained if it is known that the largest eigenvalue of A is B and the rest of the eigenvalues lie between B' and b. Under these circumstances the polynomials that satisfy

$$\max_{b \leq \lambda \leq B'} |o_k(\lambda)| = \text{a minimum}$$

and $o_k(0) = 1$ are the ones given by Zolotarev [33]. The Zolotarev polynomials are quite unwieldy.

Samokish [29] has given the approximate formula for the rate of convergence as

$$\frac{2}{\{\beta^k \cdot \frac{\beta - \alpha}{1 + \alpha\beta} + \beta^{-k} \cdot \frac{\beta - \alpha\beta}{B - \alpha}\}} \quad (8.40)$$

instead of (8.38). Here α and β are defined as

$$\beta = \frac{B' + b}{B' - b} + \sqrt{\{\frac{B' + b}{B' - b}\}^2 - 1} \quad (8.41)$$

and

$$\alpha = a + \sqrt{a^2 - 1} \quad (8.42)$$

where

$$a = \frac{2B - B' - b}{B' - b}$$

in an N dimensional quadratic problem the error tends to zero in one step

with Newton's method and within M steps with the conjugate gradient method, where M is the number of independent eigenvalues of \underline{A} . Hence it is generally stated that the conjugate gradient method yields "Quadratic convergence"

$\{\|\underline{x}_m - \underline{x}_o\| \leq C \|\underline{x}_m - \underline{x}_o\|^2\}$ in the sense that it converges in M steps for an N dimensional problem. So it seems appropriate to term this $\frac{1}{M}$ quadratic convergence since it requires M steps to achieve the effect of one step of a method with a true quadratic convergence rate.

8.2.3 THE J STEPS STEEPEST DESCENT METHOD [29]

In the J steps steepest descent method, J steps of the steepest descent are taken simultaneously rather than J individual steps one at a time. As we shall see now the J steps steepest descent has a faster rate of convergence than the J individual steepest descents, but this is not better than J steps of the conjugate gradient method.

In the J steps steepest descent method we start from an approximation \underline{x}_o and obtain the vector $\underline{x}_o + \underline{v}$, when the vector \underline{v} belongs to the subspace spanned by \underline{R}_o , \underline{AR}_o , $\underline{A}^2 \underline{R}_o$, ..., $\underline{A}^{J-1} \underline{R}_o$. Thus it can be shown that the result of one step of the J steps method of steepest descent coincides with the result of the J th approximation for the method of conjugate gradients. Hence we observe that one step of the J -step process of steepest descent does not give a worse result, in the sense of an increasing error function, than J steps of the steepest descent method. The rate of convergence of the J step steepest descent is given by

$$\|\underline{x}_k - \underline{x}_{\text{exact}}\| \leq C^J \|\underline{x}_o - \underline{x}_{\text{exact}}\| \quad (8.44)$$

where C^J is given by (8.38) with k replaced by J . In particular,

$$c^1 = \frac{B - b}{B + b}$$

$$c^2 = \frac{(B - b)^2}{(B + b)^2 + 4bB}$$

$$c^3 = \frac{(B - b)^3}{(B + b) [(B + b)^2 + 12bB]}$$

Hence it is easy to see that

$$1 > c^1 > \sqrt{c^2} > \sqrt[3]{c^3} > \dots$$

The last inequalities mean that for sufficiently large N the result of applying $\frac{N}{J}$ steps of the J steps process gives a better approximation than $\frac{N}{J-1}$ steps of the (J-1) steps processes.

Since the J steps steepest descent process is algebraically more cumbersome than J one-steps of the conjugate gradient method but yields the same estimate for the rate of convergence, we will not further discuss the J steps steepest descent method. Moreover, the J steps steepest descent method, unlike the conjugate gradient, does not terminate at the end of at most N steps.

9. ROUND-OFF ERRORS ASSOCIATED WITH ITERATIVE SCHEMES [34, 35]:

Summary:

In iterative methods the condition number of \underline{A} has very little influence on round-off error. The round-off error in iterative methods is confined to the last stage of iteration only. We show that if ΔA and ΔY are uncertainties in the matrices \underline{A} and \underline{Y} and if \underline{R}_n is the residual at the end of each iteration, then \underline{x}_n is an acceptable solution provided

$$\sum_j \Delta A^{ij} \cdot |x_n^j| + \Delta Y^i \geq |R_n^i| \quad \text{for all } i$$

Note in this case, the number of solutions for $\underline{AX} = \underline{Y}$ are infinite. The above inequality indicates that there is no need to make the residual small (to a desired accuracy) if the uncertainties in \underline{A} and \underline{Y} are large. Since in an iterative process one always computes the residual, one could terminate the iterative process when the above inequality is satisfied rather than imposing a more stringent criterion that the residuals be arbitrarily small.

9.1 ERROR ANALYSIS

In actual computation of successive approximations we have

$$\underline{x}_{n+1} = \underline{T} \underline{x}_n + \underline{w} \quad (9.1)$$

where \underline{T} is an arbitrary operator whose $\|\underline{T}\| < 1$. In general, exact determination of \underline{x}_{n+1} is impossible, since computation of the values of various matrices and numbers inevitably involves round-off errors. The only possible general assertion is that the total error in application of the operator \underline{T} does not exceed some number δ in the norm.

Thus in actual computation of successive approximations

$$\underline{x}_{n+1} = \underline{T} \underline{x}_n + \underline{w} + \underline{w}_n \quad (9.2)$$

where \underline{w}_n is an array of unknown random elements, though we have an estimate

$$\max \left| \underline{w}_n^i \right| \leq \delta \quad \text{for } i = 1, 2, \dots, N \quad (9.3)$$

where δ is a constant > 0 . The successive approximations in (9.2) may no longer converge to the solution $\underline{x}_{\text{exact}}$ of (9.1). Nevertheless we may be able to obtain an estimate of the uncertainty in the solution. We will know from (8.10)

$$\|\underline{x}_{n+1} - \underline{x}_{\text{exact}}\| \leq \lambda_1 \|\underline{x}_n - \underline{x}_{\text{exact}}\| + \delta$$

where λ_1 is the largest absolute eigenvalue of \underline{T} , and is < 1 .

Hence

$$\begin{aligned} \|\underline{x}_{n+1} - \underline{x}_{\text{exact}}\| &\leq \lambda_1^2 \|\underline{x}_{n-1} - \underline{x}_{\text{exact}}\| + \lambda_1 \delta + \delta \\ &\leq \lambda_1^{n+1} \|\underline{x}_0 - \underline{x}_{\text{exact}}\| + \lambda_1^n \delta + \lambda_1^{n-1} \delta + \dots + \delta \\ &\leq \lambda_1^{n+1} \|\underline{x}_0 - \underline{x}_{\text{exact}}\| + \frac{\delta}{1 - \lambda_1} \end{aligned} \quad (9.4)$$

Thus

$$\lim_{n \rightarrow \infty} \|\underline{x}_n - \underline{x}_{\text{exact}}\| \leq \frac{\delta}{1 - \lambda_1} \quad (9.5)$$

The formula in (9.5) may be used to obtain an approximate solution provided the required accuracy is at most $\frac{\delta}{1 - \lambda_1}$. However in most practical problems λ_1 is difficult to estimate or obtain. The amount of work required to obtain λ_1 may be almost the same as the solution of (9.1). Hence a more practical approach is taken to estimate whether an iterate \underline{X}_n is essentially an exact solution or not.

In a system of linear equations that arises from a practical problem, the elements of matrix \underline{A} and \underline{Y} may not be sharply defined. It is now assumed that all that is known about the typical element A^{ij} of matrix \underline{A} or Y^i for the matrix \underline{Y} is that they are within the following intervals. Here the subscript E denotes exact quantities

$$A_E^{ij} - \Delta A^{ij} \leq A^{ij} \leq A_E^{ij} + \Delta A^{ij} \quad (9.6)$$

$$Y_E^i - \Delta Y^i \leq Y^i \leq Y_E^i + \Delta Y^i \quad (9.7)$$

It is also assumed that ΔA^{ij} and ΔY^i are independent distinct quantities due to round-off error and one does not depend on the other. Thus

$$\underline{A}_E \underline{X}_E = \underline{Y}_E \quad (9.8)$$

and denote

$$\underline{R} = \underline{A}_E \underline{X} - \underline{Y}_E \quad (9.9)$$

We check whether

$$\sum_j \Delta A^{ij} \cdot |x^j| + \Delta Y^i \geq |R^i| \quad (9.10)$$

for $i = 1, 2, \dots, n$. As was shown by Oettli and Prager [34] the inequality (9.10) is a necessary and sufficient condition for \underline{X} to be a solution of $\underline{AX} = \underline{Y}$ under (9.6) and (9.7). So in any iteration method where the residuals are computed routinely, if for a certain residual the inequality (9.10) is

satisfied, we have obtained an excellent solution under the conditions (9.6) and (9.7). Note that the introduction of the conditions (9.6) and (9.7) does not make the solution of $\underline{A}\underline{X} = \underline{Y}$ unique. In fact, there are many solutions of $\underline{A}\underline{X} = \underline{Y}$. But, only those solutions are acceptable which satisfy the inequality (9.10). Thus, the upper and the lower bounds for a certain component of the solution \underline{X} is obtained by solving the linear programming problem [35]

$$\begin{array}{ll} \min & \left\{ \begin{array}{l} R^i - \sum_j \Delta A^{ij} \cdot |x^j| - \Delta Y^i \leq 0 \\ x^j \end{array} \right. \\ \max & \left\{ \begin{array}{l} -R^i - \sum_j \Delta A^{ij} \cdot |x^j| - \Delta Y^i \leq 0 \\ x^j \end{array} \right. \end{array}$$

for $i = 1, 2, \dots, N$ (9.11)

Thus iterative methods may be quite advantageous for large systems of matrices or for ill-conditioned matrices as compared to a direct method like Gaussian elimination. This is because cond (A) does not arise in the round-off error analysis of iterative methods.

This is the reason we have been able to solve a 7×7 system of equations when \underline{A} is a Hilbert matrix by the conjugate gradient method when Gaussian elimination has failed.

10. EXTENSION OF DIRECT AND ITERATIVE METHODS TO COMPLEX UNSYMMETRIC MATRICES

Summary

The formulas are presented for the different iterative schemes when \underline{A} is a complex unsymmetric matrix. The rate of convergence and the analysis of round-off errors are the same as obtained before, except these are now in complex arithmetic.

10.1 DIRECT METHODS

The methods described in section 2 can be used for complex unsymmetric matrices. The formulas described there can be used as they stand except now each variable is a complex number instead of real.

10.2 ITERATIVE METHODS

10.2.1 LINEAR ITERATIVE METHODS

The linear iterative methods can easily be extended to complex unsymmetric matrices. For example, in Jacobi's method, the i th element of unknown \underline{x} at $n+1$ iteration is refined in the following way

$$x_{n+1}^i = \frac{1}{a_{ii}} [y^i - \sum_{\substack{j=1 \\ j \neq i}}^N a^{ij} \cdot x_n^j] \quad \text{for } i = 1, 2, \dots, N \quad (10.1)$$

and the corresponding formula for Seidel's method is

$$x_{n+1}^i = \frac{1}{a_{ii}} [y^i - \sum_{j=i+1}^N a^{ij} x_n^j - \sum_{j=1}^{i-1} a^{ij} x_{n+1}^j] \quad (10.2)$$

10.2.2 NONLINEAR ITERATIVE METHODS

For unsymmetric complex matrices, the nonlinear iterative methods may be used on the symmetric system of equations $\underline{A}^T \underline{AX} = \underline{A}^T \underline{Y}$ instead of on the unsymmetric equations $\underline{AX} = \underline{Y}$, where T denotes the conjugate transpose of the matrix. In addition, the definition of the inner products has been redefined as shown.

10.2.2.1 METHOD OF STEEPEST DESCENT

In the method of steepest descent, the successive iterates are generated by

$$\underline{x}_{n+1} = \underline{x}_n + t_n \cdot \underline{A}^T \underline{R}_n$$

where $\underline{R}_n = \underline{A}\underline{x}_n - \underline{Y}$ and $t_n = -\frac{\langle \underline{A}^T \underline{R}_n, (\underline{A}^T \underline{x}_n)^* \rangle}{\langle \underline{A}\underline{A}^T \underline{R}_n, (\underline{A}^T \underline{x}_n)^* \rangle}$

where * denotes the complex conjugate.

10.2.2.2. CONJUGATE GRADIENT METHOD

The conjugate gradient method is now extended to the complex unsymmetric set of equations $\underline{A}\underline{x} = \underline{Y}$. We start with an initial guess \underline{x}_0 and generate

$$\underline{P}_0 = -\underline{A}^T \underline{R}_0 = -\underline{A}^T [\underline{A}\underline{x}_0 - \underline{Y}]$$

and then develop

$$\underline{x}_{n+1} = \underline{x}_n + t_n \underline{P}_n$$

where

$$t_n = -\frac{\langle \underline{A}\underline{P}_n, \underline{P}_n^* \rangle}{\langle \underline{A}\underline{P}_n, (\underline{A}\underline{P}_n)^* \rangle} = \frac{\|\underline{A}^T \underline{R}_n\|^2}{\|\underline{A}\underline{P}_n\|^2}$$

The residuals are generated as

$$\underline{R}_{n+1} = \underline{R}_n + t_n \cdot \underline{A}\underline{P}_n$$

The direction vectors are obtained iteratively as

$$\underline{P}_{n+1} = -\underline{A}^T \underline{R}_{n+1} + q_n \underline{P}_n$$

where $q_n = \frac{\langle \underline{A}\underline{P}_n, (\underline{A}\underline{A}^T \underline{R}_{n+1})^* \rangle}{\langle \underline{A}\underline{P}_n, (\underline{A}\underline{P}_n)^* \rangle} = \frac{\|\underline{A}^T \underline{R}_{n+1}\|^2}{\|\underline{A}^T \underline{R}_n\|^2}$

11. MINIMIZATION OF THE CONDITION NUMBER OF A MATRIX FOR ACCELERATING ITERATIVE METHODS AND REDUCING ROUND-OFF ERRORS IN DIRECT METHODS [36]:

Summary

We have shown in section 2 that the higher the cond $\underline{[A]}$ the greater is the amount of round-off error associated with the process. In section 8, we have shown that the higher the cond $\underline{[A]}$ the slower is the rate of convergence for the iterative schemes. Hence it would be useful to preprocess the equations $\underline{AX} = \underline{Y}$ to form $\underline{\Lambda}'\underline{X} = \underline{Y}'$ such that $\underline{\Lambda}'$ has a reduced condition number. In this section a method is outlined to reduce the condition number of \underline{A} . Also for a problem which is to be solved once, this method is impractical. This is because we need approximately as many computations to solve for cond $\underline{[A]}$ as we need to solve the original problem.

11.1 DERIVATION OF THE OPTIMUM ACCELERATION PARAMETER

In the solution of $\underline{AX} = \underline{Y}$ by any standard iterative methods we have observed that the rate of convergence depends inversely on the condition number of matrix \underline{A} . For example, for the steepest descent method the rate of convergence is proportional to $\frac{\text{cond } (\underline{A}) - 1}{\text{cond } (\underline{A}) + 1}$, where $\text{cond } (\underline{A}) \triangleq \frac{\text{largest eigenvalue of } \underline{A}}{\text{smallest eigenvalue of } \underline{A}}$.

Also for the conjugate gradient method, the rate of convergence is proportional to $\frac{\sqrt{\text{cond } (\underline{A})} - 1}{\sqrt{\text{cond } (\underline{A})} + 1}$.

Direct methods of solution of $\underline{AX} = \underline{Y}$ are also affected by $\text{cond } (\underline{A})$. This is clear from section 2.5 where the round-off error associated with the solution of $\underline{AX} = \underline{Y}$ is directly related to $\text{cond } (\underline{A})$.

In both of the examples above we see that the lower the $\text{cond } (\underline{A})$, the faster the rate of convergence of the iterative methods and the lower the round-off error for direct methods. Hence in this section we outline a procedure for the reduction of the $\text{cond } (\underline{A})$.

Again for simplicity of analysis we shall assume \underline{A} to be symmetric and definite, although this method can also be applied to unsymmetric matrices.

We now transform

$$\underline{AX} = \underline{Y} \quad (11.1)$$

to the following form

$$[\underline{I} + \omega \underline{L}]^{-1} [\underline{A}] [\underline{I} + \omega \underline{U}]^{-1} [\underline{I} + \omega \underline{U}] [\underline{X}] = [\underline{I} + \omega \underline{L}]^{-1} [\underline{Y}] \quad (11.2)$$

where ω is an acceleration parameter which is to be determined. Also

\underline{I} is the identity matrix and the matrix equations $\underline{AX} = \underline{Y}$ are so scaled that $\underline{A} = \underline{I} + \underline{L} + \underline{U}$, where \underline{L} and \underline{U} are the lower and the upper triangular matrix, respectively.

Let $\underline{D} \triangleq [\underline{I} + \omega \underline{U}] [\underline{X}] \quad (11.3)$

$$\underline{C} \triangleq [\underline{I} + \omega \underline{L}]^{-1} [\underline{Y}] \quad (11.4)$$

then (11.2) is reduced to

$$[\underline{I} + \omega \underline{L}]^{-1} [\underline{A}] [\underline{I} + \omega \underline{U}]^{-1} [\underline{D}] = [\underline{C}]$$

or $\underline{\Omega}^T \cdot \underline{A} \cdot \underline{\Omega} \cdot \underline{D} = \underline{C}$ (11.5)

It is clear that $\underline{\Omega}$ is in a form which can be readily inverted. Further if we define

$$\underline{B} = [\underline{I} + \omega \underline{U}]^{-1} [\underline{I} + \omega \underline{L}]^{-1} \underline{A} \quad (11.6)$$

then

$$[\underline{I} + \omega \underline{U}] [\underline{B}] [\underline{I} + \omega \underline{L}]^{-1} = \underline{\Omega}^T \underline{A} \underline{\Omega} \quad (11.7)$$

Hence $[\underline{B}]$ has the same eigenvalues $\lambda_i(\omega)$ for $i = 1, 2, \dots, N$ as that of $\underline{\Omega}^T \underline{A} \underline{\Omega}$, but has different eigenvectors. Thus

$$\underline{B}\underline{V} = [\underline{I} + \omega \underline{U}]^{-1} [\underline{I} + \omega \underline{L}]^{-1} [\underline{A}]\underline{V} = \lambda_i(\omega) \underline{V} \quad (11.8)$$

where \underline{V} are the eigenvectors of matrix \underline{B} . If we define

$$\underline{V}^T \underline{A} \underline{V} = \tau_i \quad (11.9)$$

$$\underline{V}^T \cdot \underline{L} \cdot \underline{U} \cdot \underline{V} = \theta_i \quad (11.10)$$

then by multiplying both sides of (11.8) by

$$[\underline{V}^T] [\underline{I} + \omega \underline{L}] [\underline{I} + \omega \underline{U}]$$

we obtain

$$\tau_i = (1 - \omega + \omega \tau_i + \omega^2 \theta_i) \lambda_i(\omega) \quad (11.11)$$

and so

$$\lambda_i(\omega) = \frac{\tau_i}{1 - \omega + \omega \tau_i + \omega^2 \theta_i} \quad (11.12)$$

Hence cond $[\underline{\Omega}^T \underline{A} \underline{\Omega}]$ is obtained as

$$\text{cond } [\underline{\Omega}^T \underline{A} \underline{\Omega}] = \frac{\lambda_1(\omega)}{\lambda_N(\omega)} = \frac{\tau_1 (1 - \omega + \omega \tau_N + \omega^2 \theta_N)}{\tau_N (1 - \omega + \omega \tau_1 + \omega^2 \theta_1)} \quad (11.13)$$

For cond $[\underline{\Omega}^T \underline{A} \underline{\Omega}]$ to have a minimum value we must require

$$\frac{d}{d\omega} \text{cond } [\underline{\Omega}^T \underline{A} \underline{\Omega}] = 0 \quad (11.14)$$

This results in a quadratic equation

$$\omega^2 (\tau_1 \theta_N - \tau_N \theta_1 + \theta_1 - \theta_N) + 2\omega (\theta_N - \theta_1) + (\tau_N - \tau_1) = 0 \quad (11.15)$$

Thus the optimum preconditioning parameter ω is obtained as

$$\omega_{\text{opt}} = \frac{\tau_N - \tau_1}{(\theta_1 - \theta_N) - ((\theta_1 - \theta_N)^2 - (\tau_N - \tau_1)(\theta_N \tau_1 - \theta_1 \tau_N) - \theta_1 \theta_N)} \quad (11.16)$$

and the minimum $\text{cond } [\underline{\Omega}^T \underline{A} \underline{\Omega}]$ is given by

$$\min \{ \text{cond } [\underline{\Omega}^T \underline{A} \underline{\Omega}] \} = \frac{\tau_1 (1 - \omega_{\text{opt}} + \omega_{\text{opt}} \tau_N + \omega_{\text{opt}}^2 \theta_N)}{\tau_N (1 - \omega_{\text{opt}} + \omega_{\text{opt}} \tau_1 + \omega_{\text{opt}}^2 \theta_1)} \quad (11.17)$$

The eigenvalues of the matrix $\underline{\Omega}^T \underline{A} \underline{\Omega}$ is bounded by the values λ_{\min} and λ_{\max} such that

$$0 < \lambda_{\min} \leq \lambda_i(\omega) \leq \lambda_{\max} \quad (i = 1, 2, \dots, N)$$

where $\lambda_{\min} = \frac{\tau_N}{(1 - \omega_{\text{opt}} + \omega_{\text{opt}} \tau_N + \omega_{\text{opt}}^2 \theta_N)}$ (11.18)

$$\lambda_{\max} = \frac{\tau_1}{(1 - \omega_{\text{opt}} + \omega_{\text{opt}} \tau_1 + \omega_{\text{opt}}^2 \theta_1)} \quad (11.19)$$

Even though simple analytical expressions are available, these are of little use for practical purposes. Only under certain conditions are analytical evaluations possible. For example when matrix \underline{A} has the following structure

$$\begin{bmatrix} I_1 & D^T \\ -D & I_2 \end{bmatrix}$$

where \underline{I}_1 & \underline{I}_2 are identity matrices and \underline{D} is a diagonal matrix, then Evans has obtained ω_{opt} as unity. When matrix \underline{A} is formed as

$$\underline{A} = \underline{I} + \underline{L} + \underline{L}^T$$

then the maximum eigenvalue of \underline{A} is minimized in $\underline{\Omega}^T \underline{A} \underline{\Omega}$ when $\omega_{opt} < 2$. Thus the cond $[\underline{A}]$ is minimized to cond $[\underline{\Omega}^T \underline{A} \underline{\Omega}]$ for $1 < \omega_{opt} < 2$. For most cases however ω_{opt} has to be determined experimentally, or by (11.16), (11.18) and (11.19). Evans has shown that there is a remarkable correlation between theoretical and experimental values of ω_{opt} obtained for a particular matrix \underline{A} .

12. CORE STORAGE REQUIRED FOR VARIOUS METHODS [1]:

The core storage required for various methods are listed in order of the amount of core storage required starting with the method requiring the least core storage (N is the rank of the matrix A)

METHOD	CORE STORAGE
Gaussian elimination	$N^2 + 2N$
Seidel's Iterative method	$N^2 + 2N$
Gaussian elimination with complete pivoting	$N^2 + 3N$
Jacobi's Iterative method	$N^2 + 3N$
Monte Carlo method	$N^2 + 3N + 14$
Method of Steepest Descent	$N^2 + 4N + 2$
Conjugate gradient method	$N^2 + 6N + 3$

It is found that Gaussian elimination with no pivoting and Seidel's method require the least amount of storage and that the conjugate gradient method requires the largest amount of storage. For complex matrices the amount of storage is doubled. For symmetric matrices, however, N^2 could be replaced by $N^2/2$.

13. WORK REQUIRED FOR VARIOUS METHODS [1]:

The number of divisions, multiplications and additions/subtractions provide a rough estimate of the efficiency of the algorithm. For each method it is possible to estimate the number of arithmetic operations as a function of N -- the order of the matrix. Such functions could be discontinuous if N is large enough that auxiliary storage is required. In the total number of arithmetic operations we have not included the timings taken for recording of intermediate results and the time taken for searching the pivotal element in Gaussian elimination.

METHOD	NUMBER OF ARITHMETIC OPERATIONS		
	Divisions ÷	Multiplications X	Additions +
Gaussian elimination	N	$\frac{N^3}{3} + N^2 - \frac{N}{3}$	$\frac{N^3}{3} + \frac{N^2}{2} - \frac{5N}{6}$ (total)
Gaussian elimination with complete pivoting			$\frac{N^3}{3} + \frac{N^2}{2} - \frac{5N}{6}$ comparisons
			in addition to the above arithmetic operations
Jacobi & Seidel	N	N^2	$N^2 - N$ per iteration
			(we could do with N divisions only once rather than per iteration at the expense of N more storage spaces.)
Steepest Descent (for unsymmetric <u>A</u>)	1	$2N^2 + 3N$	$2N^2 + 4N$ per iteration
Conjugate gradient (for unsymmetric <u>A</u>)	2	$2N^2 + 6N$	$2N^2 + 6N$ per iteration
Monte Carlo	CW	$1ZN + 8C$	$2[3N + (W + 2) C]$ (total)
			[W is the average number of steps per random walk and C is the number of random walks]

Also note that when the matrix is symmetric the number of operations is reduced by about half. Also note in the Monte Carlo method the amount of work required varies as the first power of N only. Thus the Monte Carlo method

would be quite advantageous in providing an initial guess which may be refined by the nonlinear iterative methods since they have a faster rate of convergence than linear iterative methods.

For very large values of N , an iterative method applied to a full matrix would need to converge in less than $\frac{N}{3}$ steps to bring its operations count down to that of a direct method.

For complex matrices, these operations of divisions, multiplications and additions refer to complex arithmetic operations.

14. A SPECIAL NOTE ON THE CONJUGATE GRADIENT METHOD

An iterative method called the banded matrix iterative scheme has recently been applied by Ferguson [37] to solve large electromagnetic field problems by the method of moments. The characteristic features of the method applied by Ferguson are:

- 1) The convergence of the iterative scheme is sensitive to the choice of the numbering scheme used.
- 2) Because of (1) it requires a person with certain technical background to run the program.
- 3) The rate of convergence is irregular and sometimes the solution diverges.
- 4) The banded matrix iterative scheme applied by Ferguson is basically a Jacobi type of iterative scheme and hence it converges slowly [p. 16 of ref. 37].
- 5) Finally the method needs theoretically an infinite number of steps to converge to the exact result if there is no round-off error.

An alternative scheme is proposed here to replace the banded matrix iterative scheme by the conjugate gradient method in the RADC GEMACS program. As we shall presently demonstrate, the conjugate gradient method is also capable of replacing the iterative methods in the RADC nonlinear system identification programs, too.

As we have seen from the previous sections the conjugate gradient method is a nonlinear iterative scheme, in contrast to the linear Jacobi method. Also the conjugate gradient method converges at a faster rate than that of a geometric series. Moreover it is highly insensitive to the choice of the initial guess for the solution. Since the conjugate gradient method yields an

exact result (assuming no round-off errors) in at most M steps (where M is the number of independent eigenvalues of the $N \times N$ matrix), it has the good points of both an iterative method and a direct method of solution. It has the advantage of an iterative scheme in that round-off error is limited only to the final step of the solution. It has the advantage of a direct method in that it converges in a finite number of steps.

As a first example consider a wire 3m in length and .01m in radius. The wire is charged to a potential of $4\pi\varepsilon$ volts. The objective is to find the charge distribution on the wire. A method of moments formulation has been employed and the wire is divided into a total number of 30 segments. The moment matrix formed by this problem is a typical one which often occurs in the method of moments. The results are presented in Table 8. The first three columns indicate the charge distribution on the wire obtained by the conjugate gradient method. The third column represents the charge distribution corresponding to the segment numbers appearing in column two. The first column states that this result has been obtained at the end of three iterations. The next three columns indicate the charge distribution obtained after eight iterations by the conjugate gradient method. And finally the seventh column gives the result due to Gaussian elimination. As is clear from the data presented in Table 8 the conjugate gradient method yields a result better than 1% after three iterations ($M = \frac{N}{10}$).

If for this problem the banded matrix technique is used to yield an accuracy of 1% in one iteration a bandwidth of approximately 15 may be necessary (see table 10, p. 34 of ref. 37). Hence for the same accuracy the conjugate gradient method is faster by a factor of 2.5. Also if the same problem is to be solved by the symmetric Cholesky decomposition it would have required ap-

①		1	2	3	4	5	6	7
3	1310801E+00	8	8	8	8	8	8	8
3	.1079367E+00	9	9	9	9	9	9	9
3	.9741676E-01	10	10	10	10	10	10	10
3	.9380448E-01	11	11	11	11	11	11	11
3	.9250486E-01	12	12	12	12	12	12	12
3	.8838725E-01	13	13	13	13	13	13	13
3	.8869338E-01	14	14	14	14	14	14	14
3	.8770591E-01	15	15	15	15	15	15	15
3	.8790541E-01	16	16	16	16	16	16	16
3	.8776426E-01	17	17	17	17	17	17	17
3	.8714312E-01	18	18	18	18	18	18	18
3	.8617383E-01	19	19	19	19	19	19	19
3	.8600593E-01	20	20	20	20	20	20	20
3	.8614635E-01	21	21	21	21	21	21	21
3	.8534630E-01	22	22	22	22	22	22	22
3	.8631968E-01	23	23	23	23	23	23	23
3	.8616120E-01	24	24	24	24	24	24	24
3	.8592218E-01	25	25	25	25	25	25	25
3	.8718151E-01	26	26	26	26	26	26	26
3	.8758122E-01	27	27	27	27	27	27	27
3	.8841532E-01	28	28	28	28	28	28	28
3	.8897913E-01	29	29	29	29	29	29	29
3	.9207904E-01	30	30	30	30	30	30	30
3	.9305668E-01	31	31	31	31	31	31	31
3	.9872329E-01	32	32	32	32	32	32	32
3	.1081076E+00	33	33	33	33	33	33	33
3	.1315722E+00	34	34	34	34	34	34	34

Conjugate gradient method
after three iterations

Conjugate gradient method
after eight iterations

Gaussian
elimination

Table 5: Comparison of the different solutions for the heat wire

proximately $\frac{N^3}{6}$ multiplications. The conjugate gradient method required approximately $3N^2$ multiplications as compared to $5N^2$ for Gaussian elimination.

Also note that an essentially exact result has been obtained (accuracy better than 10^{-5} in the residuals) after only eight iterations.

As a second example, consider the same problem as above but now the wire is 25m long. So this time A is 100×100 matrix. Again we obtained an essentially exact result (better than 10^{-5} in the residuals) after only nine iterations. This implies that in these type of problems the number of independent eigenvalues is approximately eight or nine. Note that the number of independent distinguishable eigenvalues does not increase as the order of the system is increased considerably. This is an interesting property of diagonally dominant matrices which could easily be exploited by the conjugate gradient method.

As a third example, consider A as a 20×20 Hilbert matrix and y is chosen in such a way that the solution vector has components 1 to 20. The problem then is to find x given A and y. The philosophy behind choosing A to be a Hilbert matrix is that nearly singular matrices are often encountered in a system identification problem. So if the conjugate gradient method can efficiently solve such an ill-conditioned problem, then this method may easily be applicable to system-identification problems. The results obtained by two different methods are shown in Table 9.

It is clear from Table 9 that the conjugate gradient method yields good results at the end of eight steps. The largest error is only 2.25%. The Gaussian elimination method for the same problem completely breaks down.

(Note. The Hilbert matrix is extremely ill-conditioned. The condition number of a 20×20 Hilbert matrix is of the order of $e^{3.5N} = 2.5 \times 10^{30}$ from

<u>Gaussian elimination</u>	<u>Exact solution</u>	<u>Conjugate gradient at the end of 8 steps</u>
.9999954	1	1.000289
2.000349	2	1.990388
2.978199	3	3.056398
4.103616	4	3.909776*
4.355928	5	4.981514
55.54727	6	6.056422
-20.17007	7	7.066274
-391.4351	8	8.030005
1050.932	9	8.982147
-397.4495	10	9.947065
212.3952	11	10.93533
1407.415	12	11.94698
-1800.392	13	12.97573
-682.3352	14	14.01228
2770.054	15	15.04653
-1692.660	16	16.06884
637.3160	17	17.07074
559.0285	18	18.04519
78.80465	19	18.98660
97.65299	20	19.8907

Table 9: Comparison of Gaussian elimination and conjugate gradient method for the solution vector $\underline{x} = [\underline{A}]^{-1} \underline{y}$.

*The largest error is about 2.25%

[38].

As a final example consider the solution of the two components of the current density on a 1λ square metal plate irradiated by a plane wave. When the total number of unknowns for the complex current is 71, we have to solve a 71×71 matrix equation. The total time taken for the solution of the complete problem utilizing various techniques is as follows:

Gaussian elimination: 27 sec. (CPU time)
Conjugate gradient method: 30 sec. (CPU time)
(with 1% accuracy in the residual)

Observe that the conjugate gradient method is quite inefficient in this case. However, as the dimension of the problem is increased from 71 to 180, the time required by various methods to solve the complete problem is as follows:

Gaussian elimination: 500 sec. (CPU time)
Conjugate gradient method:
for 10^{-2} accuracy in the residual: 220 sec. (CPU time)
for 10^{-3} accuracy in the residual: 290 sec. (CPU time)
for 10^{-4} accuracy in the residual: 390 sec. (CPU time)
for 10^{-5} accuracy in the residual: 520 sec. (CPU time)

So for large systems of equations the conjugate gradient method may prove to be quite useful, especially if one is interested in obtaining an accuracy of 10^{-3} to 10^{-4} in the solutions.

In summary, it is argued that the application of the conjugate gradient method to the analysis of large bodies by method of moments would yield stable, reliable, consistent and accurate results faster than any methods currently used to obtain a solution. The same is true for problems in system identification. However there may be some build-up of the round-off error if the residuals are computed iteratively by (7.10) rather than directly from $\underline{AX}_k = \underline{Y}$, which would be more time-consuming. At this point it is not known how serious this problem will be for our problems of interest.

15. SUMMARY AND CONCLUSIONS

Of all the stationary iterative schemes surveyed the conjugate gradient method has shown great promise as a possible candidate to replace the banded matrix iterative scheme in the GEMACS program. This is because the conjugate gradient method not only yields the exact solution theoretically at the end of a finite number of steps but also has the fastest rate of convergence.

The next step of the program should be to develop computer programs for the various methods and verify experimentally the theoretical results that have been presented in this report.

16. REFERENCES:

1. Westlake, J., "A Handbook of Numerical Matrix Inversion and Solution of Linear Equations", John Wiley & Sons, Inc., N.Y., 1968.
2. Forsythe, G., and C. B. Moler, "Computer Solution of Linear Algebraic Systems", Prentice Hall, Englewood Cliffs, N.J., 1967.
3. Fadeev D. K. and V. N. Fadeeva, "Computational Methods of Linear Algebra", (translated by R. C. Williams from Russian) W H. Freeman and Co., San Francisco, 1963.
4. Klyuyev, V. V. and N. I. Kokovkin-Scherbak, "On the Minimization of the Number of Arithmetic Operations for the Solution of Linear Algebraic Systems of Equations" (translated by G. J. Tee), Computer Science Department, T.R.-CS24, Stanford University, 1965.
5. Winograd, S., "On the Number of Multiplications Required to Compute Certain Functions", Proc. Nat. Acad. Sci., 58, 1967, pp. 1840-42.
6. Winograd, S., "A New Algorithm for Inner Product", IEEE Trans. on Computers, 17, 1968, pp. 693-694.
7. Strassen, Volker, "Gaussian Elimination is Not Optimal", Numer. Math., Vol. 13, 1969, pp. 354-365.
8. Wilkinson, J. H., "Rounding Errors in Algebraic Processes", Prentice Hall, Englewood Cliffs, N.J., 1963.
9. Schwarz, H. R., H. Rutishauser and E. Stiefel, "Numerical Analysis of Symmetric Matrices" Prentice Hall, Englewood Cliffs, N.J., 1973.
10. Hestenes, Magnus R., "Applications of the Theory of Quadratic Forms in Hilbert Space to the Calculus of Variations", Pacific J. Math., 1, 1951, pp. 525-581.
11. Hayes, R. M., "Iterative Methods of Solving Linear problems in Hilbert Space", in O. Taussky (ed.), Contributions to the solution of systems of linear equations and the determination of eigenvalues, Nat. Bur. Standards Appl. Math. Ser., Vol. 39, 1954, pp. 71-104.
12. Forsythe, G. E., "Solving Linear Equations Can Be Interesting", Bull. of Am. Math. Society, 1953, pp. 299-329.
13. Forsythe, G. E., "Theory of Selected Methods of Finite Matrix Inversion and Decomposition", U.S. Dept. of Commerce, INA 52-5.
14. Kahan, W., "Gauss Seidel Methods for Solving Large Systems of Linear Equations", Ph.D. dissertation, University of Toronto, 1958.
15. Young, D., "On the solution of Linear Systems by Iterations", in Am. Math. Soc. Numerical Analysis Proceedings of Symposia in Applied Mathematics, J. H. Curtiss (ed.), McGraw Hill, N.Y., 1956, vol. 6, pp. 283-298.

16. Young, D., "On the Solution of Large Systems of Linear Algebraic Equations With Sparse Positive Definite Matrices", in Numerical Solution of Nonlinear Algebraic Equations by G. D. Byrne (ed.), Academic Press, 1974.
17. Forsythe, G. E. and R. A. Liebler, "Matrix Inversion by a Monte Carlo Methods": MTAC, Vol 4, 1950, pp. 127-129.
18. Curtiss, J. H. "Monte Carlo Methods for Iteration of Linear Operators", Journal of Math. and Phys., 1953, pp. 209-232.
19. Curtiss, J.H., "A Theoretical Comparision of the Efficiencies of Two Classical Methods and a Monte Carlo Method for Computing One Component of the Solution of a Set of Linear Algebraic Equations", in H. A. Meyer (ed.) Symposium on Monte Carlo Methods, Wiley, New York, 1961.
20. Ralston, A., and H. Wilf. "Mathematical Methods for Digital Computers", Wiley, New York, 1960. Vol. I.
21. Wasow, W. R., "A Note on the Inversion of Matrices by Random Walks", MTAC, Vol. 6, 1952, pp. 78-81.
22. Kantorovich, L. V., "Functional Analysis and Applied Mathematics", Uspekhi Matematicheskikh Nauk, Vol. III, No. 6, pp. 89-185, 1948.
23. Engeli, M., Th. Ginsburg, H. Rutishauser and E. Stiefel, "Refined Iterative Methods for the Computation of the Solution and Eigenvalues of Self-adjoint Boundary Value Problems", Mitteilungen aus dem Institut für angewandte Mathematik, Nr. 8, 1959.
24. Daniel, J. W., "The Conjugate Gradient Method for Linear and Nonlinear Operator Equations", SIAM J. Numer. Anal., Vol. 4, No. 1, 1967, pp. 10-26.
25. Hestenes, M. and E. Stiefel, "Method of Conjugate Gradients for Solving Linear Systems", J. Res. Nat. Bur. Standards, 49 (1952), pp. 409-436.
26. Daniel, J. W., "The Conjugate Gradient Method for Linear and Nonlinear Operator Equations", Ph.D. Thesis, Stanford University, 1965.
27. Kaniel, S., "Estimates for Some Computational Techniques in Linear Algebra", Math. Comp., 20 (1966), pp. 369-378.
28. Meinardus, G., "Über eine Verallgemeinerung einer Ungleichung Von L. V. Kantorovich", Numer. Math., V. 5, 1963, pp. 14-23.
29. B. A. Somokish, "An Investigation of the Rate of Convergence for the Method of Steepest Descent", Uspekhi Matem. Nauk, 1957, Vol. 12, No. 1, pp. 238-240.

30. Nashed, M. Z., "Steepest Descent for Singular Linear Operator Equations", SIAM J. Numer. Anal., Vol. 7, No. 3, 1970, pp. 358-362.
31. Akaike, H., "On a Successive Transformation of the Probability Distribution and its Application to the Analysis of the Optimum Gradient Method", Ann. Inst. Statistics Math., Tokyo, Vol. 11, (1959), pp. 1-16.
32. Kammerer, W. J. and M. Z. Nashed, "On the Convergence of the Conjugate Gradient Method for Singular Linear Operator Equations", SIAM J. Numer. Anal., Vol. 9, No. 1, 1972, pp. 165-181.
33. Levy, R., "Generalized Rational Function Approximation in Finite Intervals Using Zolotarev Functions", IEEE Trans. on Microwave Theory & Tech., Vol. MTT-18, 1970, pp. 1052-1064.
34. Oettli, W. and W. Prager, "Compatibility of Approximate Solution of Linear Equations with Given Error Bounds for Coefficients and Right Hand Sides", Num. Math., 6, 1964, pp. 405-409.
35. Oettli, W., "On the Solution Set of a Linear System with Inaccurate Coefficients", J. SIAM Numer. Anal., Ser. B., Vol. 2, No. 1, 1965, pp. 115-119.
36. Evans, D. J., "The Use of Preconditioning in Iterative Methods for Solving Linear Equations with Symmetric Positive Definite Matrices", J. Inst. Math. Applies., 4, 1968, pp. 295-314.
37. Ferguson, T. R., "The EMCAP Iterative Techniques in the Method of Moments", RADC-TR-75-121, May 1975.
38. Marcus, M., "Basic Theorems in Matrix Theory", U.S. Dept. of Commerce, NBS, Appl. Math. Ser. 57, 1959.



MISSION of Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C³I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.